Calhoun: The NPS Institutional Archive

DSpace Repository

Theses and Dissertations 1. Thesis and Dissertation Collection, all items

1965

A study of some software parameters in time-sharing systems.

Grimes, Fred M.; Otto, Ronald E.

Monterey, California: U.S. Naval Postgraduate School

http://hdl.handle.net/10945/13182

This publication is a work of the U.S. Government as defined in Title 17, United States Code, Section 101. Copyright protection is not available for this work in the United States.

Downloaded from NPS Archive: Calhoun

Calhoun is the Naval Postgraduate School's public access digital repository for research materials and institutional publications created by the NPS community. Calhoun is named for Professor of Mathematics Guy K. Calhoun, NPS's first appointed -- and published -- scholarly author.

Dudley Knox Library / Naval Postgraduate School
411 Dyer Road / 1 University Circle
Monterey, California USA 93943

http://www.nps.edu/library

NPS ARCHIVE
1965
GRIMES, F.

A STUDY OF SOME SOFTWARE PARAMETERS IN TIME-SHARING SYSTEMS

FRED M. GRIMES
and
RONALD E. OTTO

DLEY KNOX LIBRARY
AL POSTGRADUATE SCHOOL
REY CA 93943-5101

A STUDY OF SOME SOFTWARE PARAMETERS IN TIME-SHARING SYSTEMS

* * * * *

Fred M. Grimes

and

Ronald E. Otto

A STUDY OF SOME SOFTWARE PARAMETERS IN TIME-SHARING SYSTEMS

by

Fred M. Grimes

Lieutenant Commander, Supply Corps, United States Navy

and

Ronald E. Otto

Lieutenant, Supply Corps, United States Navy

Submitted in partial fulfillment of
the requirements for the degree of

MASTER OF SCIENCE
IN
MANAGEMENT (DATA PROCESSING)

United States Naval Postgraduate School
Monterey, California

1965

NPS Archive
1965
Grimes, F.

Library
U. S. Naval Postgraduate School
Monterey, California

DUDLEY KNOX LIBRARY
NAVAL POSTGRADUATE SCHOOL
MONTEREY CA 93943-5101

A STUDY OF SOME SOFTWARE PARAMETERS IN TIME-SHARING SYSTEMS

by

Fred M. Grimes

and

Ronald E. Otto

This work is accepted as fulfilling

the thesis requirements for the degree of

MASTER OF SCIENCE

IN

MANAGEMENT (DATA PROCESSING)

from the

United States Naval Postgraduate School

ABSTRACT

A review is made of some existing time-sharing computer systems and an exploration of various software characteristics is conducted. This investigation is conducted using a computer program with which a typical time-sharing system can be simulated. The basic system parameters examined are: 1) the method of determining the quantum time for each user per response cycle, 2) the length of the desired response cycle, 3) the number of remote stations permitted and 4) the maximum number of users permitted in the queue at one time. The results of these simulation runs are presented. The effects of the various parameters upon the average response cycle time, the average number in the queue awaiting service, the average length of time a user is in the queue and the computational efficiency plus other characteristics of the system are discussed.

The authors wish to express their appreciation to Professor Douglas G. Williams for his sound advice and guidance during this investigation.

TABLE OF CONTENTS

Section	Title	Page
1.0	Introduction	1
2.0	Review of Time-Sharing Systems	4
	2.1 System Development Corporation	7
	2.2 Massachusetts Institute of Technology Compatible Time-Sharing System	13
	2.3 Rand Corporation's JOSS	16
	2.4 Dartmouth Time-Sharing System	18
	2.5 Other Time-Sharing Systems	21
	2.6 Hardware Advancements in Time-Sharing Systems	23
3.0	Investigation and Results of Varying Some Basic Parameters of a Time-Sharing System	27
	3.1 A Time-Sharing System and Its Basic Inputs	28
	3.2 Investigation of Some Methods of Determining the Quantum Time	35
	3.3 Effects of Varying Response Cycle Time	40
	3.4 Investigation of Varying the Number of Remote Stations in the Time-Sharing System	41
	3.5 Effects of Varying the Maximum Queue Size and the Number of Stations	45
4.0	Conclusions and Recommendations	48
	Bibliography	52

APPENDICES

I.	Program SIM - Description	54
II.	Flowchart of Program SIM	58
III.	Program SIM	63
IV.	Glossary of FORTRAN Names Used in Program SIM	84
V.	Results of Simulations	90

1.0 Introduction

That Automatic Data Processing Equipment (ADPE) is an integral part of business and scientific operations is a well accepted and immutable fact. No longer is it necessary for computer salesmen to beat their drums to draw attention to the fact that ADPE can perform certain operations with amazing speed and accuracy. This equipment is now accepted as part of the modern way of life.

Since the hurdle of acceptability has been achieved, the frontiersmen of the computer field are scanning the technological horizons looking for better ways to make better use of ADP equipment. In the past fifteen years great strides have been made in equipment usage. In the early nineteen fifties, the problems encountered and overcome were the construction and maintenance of hardware. In the mid-nineteen fifties software problems were reduced with the development of compilers which reduced the language burden of the user. In the early nineteen sixties, the computer frontiersmen started scratching the surface of a third major modification of ADPE: - the improvement of man-machine interaction by a process called time-sharing. Much exploratory work is currently being done in this field of time-sharing of Central Processing Units.

Currently, jobs are "batch-processed" on computers. In "batch-processing" a group of jobs is mounted on a single tape and fed sequentially to a computer. Each job is loaded into core memory and stays there until it is completely processed. Therefore if twenty jobs were in a batch, the whole batch would generally have to be processed before the results of any one job could be printed out. This method requires all users, regardless

of the total computational demands of their programs, to wait for output until all jobs in the batch are processed. The disadvantages of this method can be readily appreciated by a person who has a small program and desires quick service.

Before presenting the time-sharing method of processing jobs, it is best to give a more exact interpretation of the term time-sharing. Time-sharing can mean using different parts of the hardware at the same time for different tasks, or it can mean several persons making use of the computer at the same time. The first meaning, more correctly called multiprogramming, is oriented towards hardware efficiency. Whereas, the second meaning, which will be used in this paper and more correctly called time-sharing, is primarily concerned with the efficiency of persons trying to use a computer (1).

In a time-sharing system the processing of a job is different from that of "job-batching." The central processing unit is made available simultaneously to many users in a manner somewhat similar to a telephone exchange. Each user would be able to use a console or a typewriter, from a remote station and at his own pace communicate with the computer without concern for other stations which might be using the computer. In this scheme, each job is loaded into external storage, such as a disc file or a drum file, as the jobs arrive for processing. A scheduler program will then transfer each job from the external storage into main memory and pass control to the job for a certain period of time called the quantum. At the end of the quantum of time, the scheduler program takes control again, dumps the program from core back into external storage, loads a new job into core, and passes

control to it for a quantum of time. The quantum of time can be determined in a number of different ways. Most algorithms for quantum determination limit the time per program so that the response cycle time, i.e., the time interval between the beginning of a user's quantum of compute time to the beginning of his next quantum of compute time, is kept below some maximum level. For optimum user satisfaction, a response cycle should not be longer than comfortable human reaction time, of the order of ten seconds (2). Therefore if twenty stations were active, the maximum quantum time would be 500 milliseconds. It should be noted that this 500 milliseconds is not pure compute time but includes the overhead required to swap programs in or out of the core.

While fresh looks and new ideas are being developed for time-sharing systems, the basic idea is not new (1). Early relay computers of the Bell Telephone Laboratories were capable of being operated by several different users from distant stations. This took place in 1940 and while only one user could control the computer at one time it was considered desirable to have remote access to these computers as problem solving conveniences.

The purpose of this paper is to review the current major time-sharing systems and to investigate the effects of varying some of the papameters which affect a time-sharing system. The method of computer simulation will be used to develop the statistics for the analyses. It is hoped that the results of these analyses will shed more light on the problem of improving the utilization of Automatic Data Processing Equipment.

2.0 Review of Time-Sharing Systems

Time-sharing systems can be divided into three broad categories. The first system can be called a "General-Purpose System" and is the one generally referred to when one speaks of time-sharing systems. A general-purpose system attempts to provide the full range of equipment utilization which the user would normally have if he were the sole user. The user would expect:

(1) to have full use of the computer and all peripheral equipment.

(2) to be able to use any programming language.

(3) to use, with little or no change, programs and subroutines which may have been originally written for other systems.

(4) to wait for results or have the computation continue either in his absence or while he uses the console for a different task.

(5) to communicate fully with the computer both via a conventional typewriter and through some graphical device, such as a cathode ray tube with a light pen attachment or some equivalent device.

(6) to store large amounts of data and program material within the system so that he can have ready access to it.

(7) to have use of a large library of service routine and "debugging aids." (3)

The latter use is most important. Today the computer is used to solve more and more complex problems. These require that larger and more complicated programs be written to take advantage of the larger and faster computers. As more complex programs are written, more programmers' errors are made and these errors take longer to diagnose and debug. If a "batch-processing" system is being used each program bug takes several hours, and perhaps all

day, to correct. An alternative debugging procedure would be to turn the computer over to the programmer for a period of time while he tests his program. This alternative results in gross inefficiency in the use of the computer. A better and more economical alternative would be to make the computer available on a time-sharing basis with a host of debugging aids. The programmer could then use his quantum time to rapidly locate program errors and make corrections without tying up the equipment.

A typical method which the programmer might use to debug his program with a time-sharing system would be as follows. The programmer would write a sub-program in a compiler language and would want to incorporate this new sub-program into his set of programs already developed and kept in the computer center's central library, probably on a disc file. The programmer, from his remote station, would identify himself and start typing in his input using a simple command. He types in his sub-program for storage with his other sub-programs in the central file. If any typing or logic errors in his new file were noted, he can correct the file by using a special command. If the programmer is not proficient in typing, he can employ the help of a trained keypunch operator. By the use of another command the programmer can compile his sub-program, and if no diagnostics appear, he can prepare to test it. Using another command, such as "LOAD," he can have his program loaded into core when it comes his turn for computer time. The newly compiled sub-program will be converted into binary form, will contain any sub-programs previously prepared, and will draw from the computer center's library as many library sub-routines as he wishes. If loading is successful,

and does not require re-entry for the addition of missing sub-programs, another command initiates his program. The programmer notes any results he may get and can stop the program to inspect the status of various locations and variables within the set of sub-programs loaded. The programmer can then check his program, locate and correct the errors, and continue in this manner until he wishes to terminate his session by another command. After he has debugged his program he can obtain production output in the same manner. In this latter case the quantum time will be used to compute his results.

A second type of time-sharing system is more restrictive and is called a "Dedicated System" or a "One Language System." In this system the user is constrained to the use of one single programming language which may be FORTRAN or even a simpler language, such as JOSS which is used in the Rand Joss Time-Sharing System. Every user of the system has to use that same language.

The third class of time-sharing system is even more restrictive in that it deals with restrictive procedures, a common library facility, or a common data base. Two examples of this type of system are the SAGE Air Defense System and the SABRE airline reservation system. In this "Common-Data-Base System" the user can only ask certain types of questions for which the responses are already in the system, or provide data in the form of answers to questions which the system itself may ask. (3)

2.1 System Development Corporation (SDC) General Purpose Time-Sharing System

The time-sharing system which approaches the "General Purpose System" mentioned above is the Time-Sharing System (TSS) at SDC. It is perhaps the most versatile of current time-sharing systems. The system is located in the SDC Command Research Laboratory which is a facility sponsored by the Advanced Research Projects Agency. The laboratory's mission is to support studies in the effective application of automated information processing to military command and control systems.

Figure 2.1.1 is an overall diagram of the communications and digital equipment used in the laboratory. The central processing unit is a large and very fast machine manufactured by IBM for the military called the AN/FSQ-32. The major real-time, input-out processor for on-line devices is the PDP-1 computer manufactured by the Digital Equipment Corporation. Figure 2.1.2 is a more detailed schematic of the system configuration and Figure 2.1.3 lists the characteristics of the system.

The Q-32 executive program occupies 16 K words of core memory leaving the remaining 48 K for users' programs. This executive program includes routines that perform error recovery and input-output, allow on-line editing and debugging, assign storage, schedule users' programs, and interpret users' commands. The PDP-1 acts as a buffer and its executive is concerned primarily with monitoring the flow of information to and from remote devices, including control of real-time, input-output devices, code conversation, and communication switching.

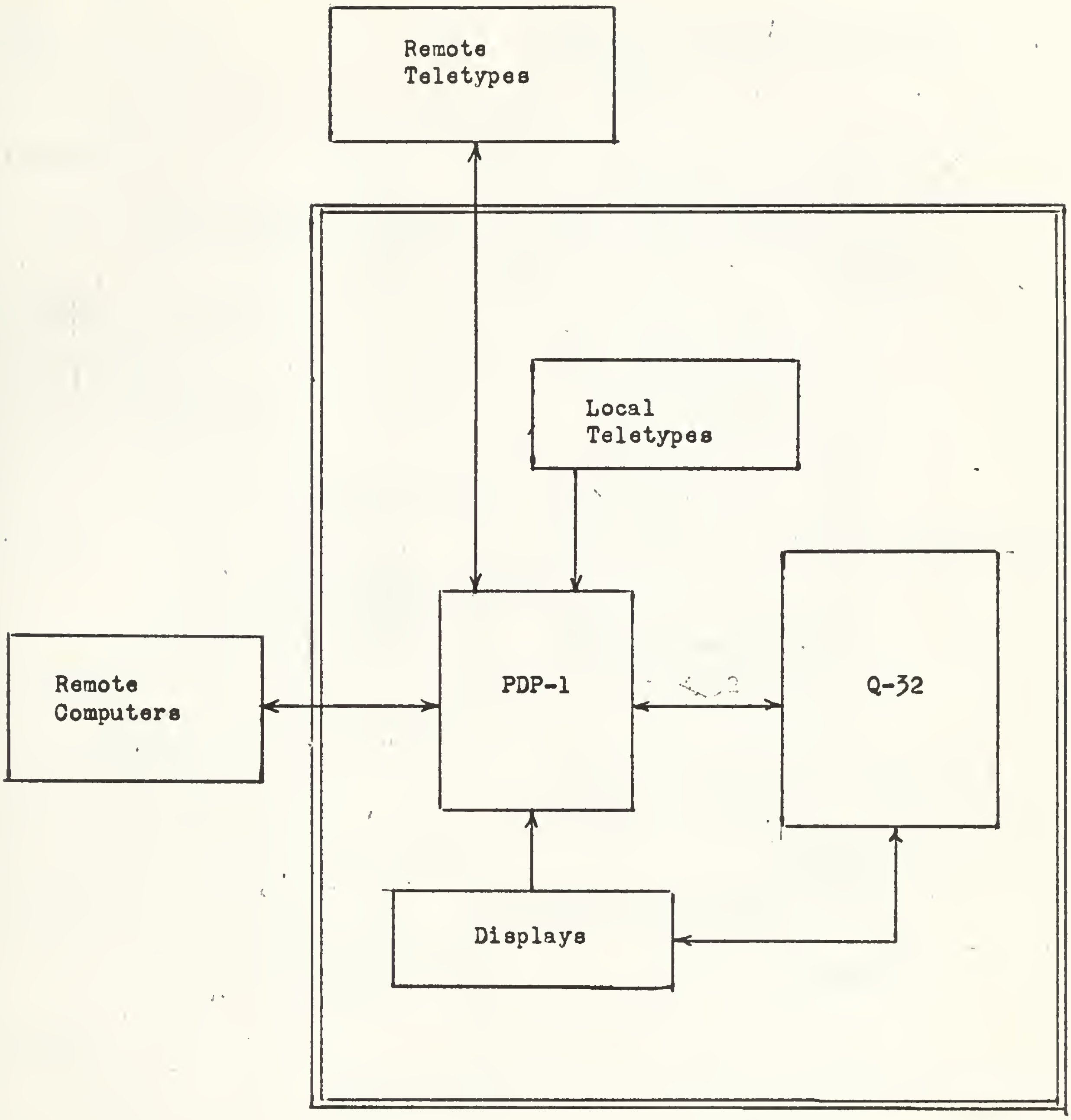


Figure 2.1.1. SDC On-Line Equipment Configuration

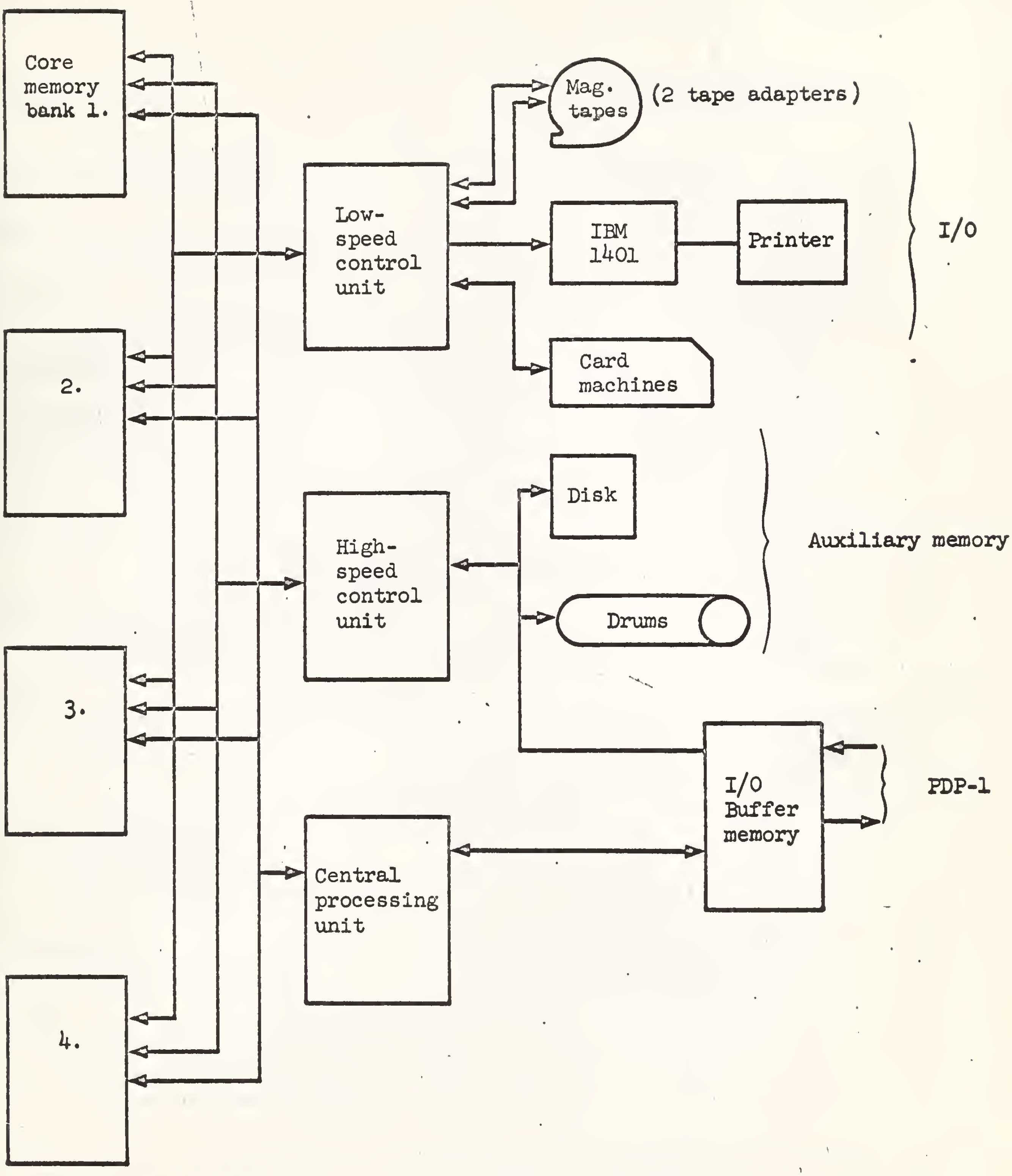


Figure 2.1.2. SDC Equipment Configuration Diagram

COMPONENT	NUMBER	CAPACITY/SPEED	TOTAL
AN/FSQ-32 COMPUTER			
Main Core Memory	4	16,384 words	65,536 words
• Cycle time 2.5 μsec.			
• 48-bit word			
Input Core Memory (Buffer)	1	16,384 words	16,384 words
• Cycle time 2.5 μsec.			
Drum	3	139,264 words	417,792 words
• Access time 11.5 ms.			
• Word transfer rate 2.75 μsec.			
Disc File	16 discs	262,144 words	4,194,304 words
• Access time 140 ms.			
• Word transfer rate 12 μsec.			
Tape Drives (729-IV)	16	112½ ips	
Card Reader	1	250 cpm	
Card Punch	1	100 cpm	
Printer	1	150 lpm	
Typewriter	2	100 wpm	
ASSOCIATED COMPUTERS (ON/OFF-LINE)			
PDP-1			
• Shares input core memory of Q-32			32K words
• Cycle time 5 μsec.			
• 18-bit word main core memory	1	4K words	4K words
1401-D			
• Core memory	1	4K char.	4K char.
• Printer	1	600 lpm	
• Tape drives (729-IV)	2	112½ ips	
I/O DEVICES			
Teletypes and Typewriters			
• Model 33 Teletypes	16	100 wpm	
• Model 28 Teletypes	8	100 wpm	
• TWX data sets (remote users)	6		
• Soroban typewriters	3	100 wpm	
Display Consoles	6	2K char. max. (per console)	
• Light pens			
• Vector-generator capability			
Telephones			
• Links for simultaneous conversations	6		
• Phones	35		
• Recording by PDP-1 (who called, when, how long)			

Figure 2.1.3. SDC Hardware Characteristics

Users communicate with the computer in the manner described earlier. There are 35 local and remote teletypewriter and typewriter stations and, by design, the system responds to any interrogation within two seconds. SDC considers a two second response highly desirable, since it is well below the ten-second maximum response cycle mentioned previously. The executive automatically retrieves each requested program from disc file, or manually from the back-up tape library, and stores the program on drum when space is available. Currently the drum can hold 400 K words, which is nine user core loads of 48 K which is the maximum allowable size of a user's program.

The program is then transferred into main memory and allotted a quantum of time computed by the scheduler program. The scheduler handles two queues, one having a one-second response time, the other longer. The executive routines can handle LISP, IPL, JOVIAL, SLIP, and SCAMP languages.

In order to meet a 10 second response time, the SDC scheduling program takes 7% of the TSS executive program space and 25% of the executive processing time to schedule users' programs. By juggling the quantum variables, the TSS can provide a two-second response cycle. This response time can be achieved by using a quantum of 400 milliseconds for an average of 12 users in the system. (2)

SDC empirical data show that the computer operating time divides roughly as follows: (4)

TYPE OF LOADING	USER PROGRAM OPERATION	SWAPPING	INPUT-OUTPUT AND OVERHEAD
HEAVY AND MODERATE	50%	35%	15%
LIGHT	20%	60%	20%

Schwartz, et al. (5) state that significant improvement in this system responsiveness is possible with dynamic relocation hardware (see Section 2.6) since it would be possible with such equipment to allow executive operation and swapping to overlap for many small programs.

2.2 Massachusetts Institute of Technology Compatible Time-Sharing System (CTSS) (1)

One of the earliest groups which worked in the computer time-sharing area was the M.I.T. Computation Center. Some of the present day work in time-sharing traces its beginning to work and papers published by computation center personnel. (6) The CTSS is basically a system which will allow the development of time-sharing while continuing to allow more conventional background systems to operate.

The central processing unit for this system was originally an IBM 7090 but was converted to a 7094 Model I in 1963. Figure 2.2.1 is a system diagram.

CTSS is a general purpose programming system which allows a new form of computer operation to evolve but yet allows older pre-time-sharing programming systems to continue to be operated. The remote consoles are of several varieties, which include, cathode ray tubes but are mostly electric typewriters. Each console user controls the computer by issuing standard commands one at a time. The commands allow convenient performance for most of the routine programming operations such as input, translation, loading, execution, stopping and inspection of programs. Although the commands are in fixed format, no loss of generality is suffered since a command can be used to start a sub-program with its own control language level. The system uses FAP, MAD, FORTRAN, SNOBOL, COMIT, GPSS, AED, OPL, DYNAMO and LISP programming languages.

The console users form what is called the foreground system, with computation being performed for the active console users in variable length bursts on a 'round-robin' basis in accordance

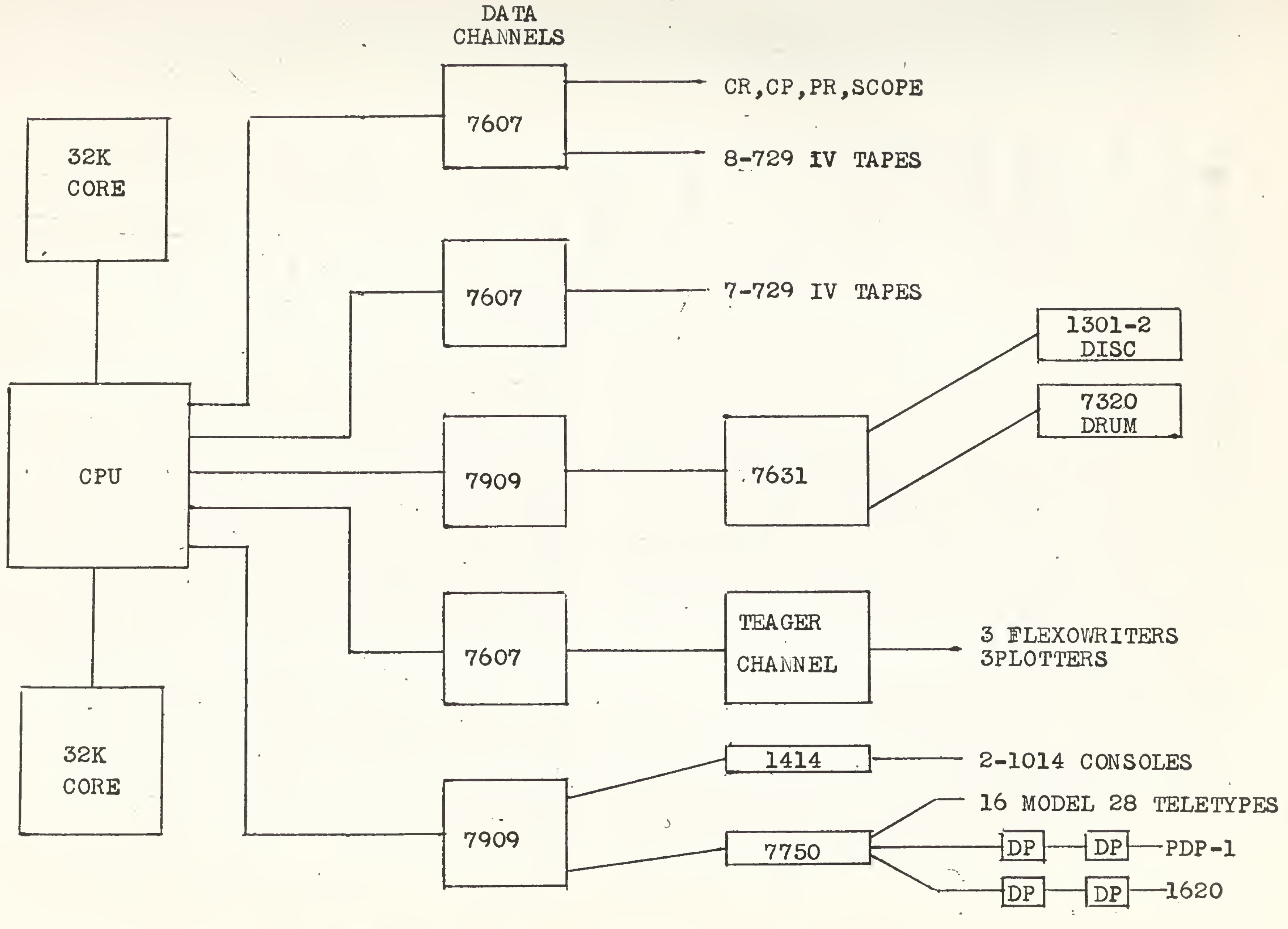


Figure 2.2.1 MIT Equipment Configuration for Time-Sharing System

with a multi-level scheduling algorithm (1). The background system is a conventional programming system which operates when the foreground system is inactive but could be also scheduled for a greater portion of computer time. The entire operation of the computer is under the control of a supervisor program which permanently resides in the 32,768 word A-bank of core memory. User programs are either kept in the 32,768 word B-bank of core memory, or swapped in and out of the external memory (disc or drum) as needed. External memory consists of two disc modules (4.6 million words each) and a drum (.2 million words). User programs, semi-permanent storage of their active programs, and data files are on the disc files. Facilities are available for cards and magnetic tapes as long-time and back-up storage devices.

In this sytem more than one program can be stored in memory at one time. At the end of each program operation time, a fixed limit of 200 milliseconds, the supervisor determines which user is to be run next. The supervisor program must then determine whether the program or programs currently in core must be dumped, in part or entirely, to make room in core for the next user. The next user must be retrieved from secondary storage together with the proper machine conditions. The computer has special modifications for memory protection of the programs in core and for relocation. The time savings of having programs already in core without swapping them unnecessarily in or out of core can be readily appreciated.

2.3 Rand Corporation's JOSS (7)

Perhaps one of the simplest and yet widely used time sharing systems is the JOSS (Johnniac Open Shop System) system at Rand Corporation, Santa Monica, California. The system was designed to provide a personal computing service tailored specifically for the needs of the scientists and engineers at Rand. The system is described as a combination desk calculator/ stored program computer not designed for production processing.

The central processing unit is a JOHNNIAC computer designed by Rand in 1950-51. The computer is accessed by ten remote typewriter (IBM 868 with a modified character set) consoles. Communication from the remote consoles to the CPU is over telephone lines. A multiple typewriter communication system mediates between the remote consoles and the central processor. Although the system has severe constraints on speed and size of program, it is in daily use and is said to provide a valuable service for computational needs of the Rand staff. The system can be classified as a "Dedicated System."

The JOHNNIAC computer has only a 4096 word memory, a slow 12,288-word drum, slow copy-logic for card input-output and printing, no tapes and a very austere order code. A special purpose buffering system was built to process characters within messages and to monitor the remote stations.

The executive program for JOSS takes about 6000 words, the low frequency portions residing on drum and overlaying each other when called into core for execution. A large portion of the executive program resides permanently in core thereby reducing core space available for users. The drum plays a major role in

the system. It is divided into three sections and accessed by a relatively slow-moving head. The average program swap time (i.e., the time required to write one user's block of information out onto drum and read a second user's block into core for processing) is quite long at about half a second.

The multiple typewriter communication system controls the state of all ten remote consoles. When a light flashes on at the remote console, it indicates that the JOSS system is available for that station's input. As soon as communication is established with JOSS, JOSS controls the typewriter, accepting input and printing output on it. The executive program controls the time quantum and when a user's program is loaded. Each user's program is stored on the drum except when actually in core. The system operates on a priority basis; each user is given a complete quantum of two seconds each. Under a typical load JOSS is said to respond to simple requests in a fraction of a second but sometimes takes as long as three seconds. The system language is constrained to the specially developed language "JOSS."

2.4 Dartmouth Time Sharing System (8)

A good example of a large scale "Dedicated System" is that in operation at the Computation Center at Dartmouth College. The purpose of this system is to teach computer programming to a large number of undergraduate students.

As shown in Figure 2.4.1 this system consists of two computers. The General Electric Datanet-30 is used as the remote console controller and also to hold the master executive program. The main computer, a General Electric GE-235, performs the computation and is controlled by the smaller computer. While there is a direct line between the computers for control purposes, the data and information is transferred via a disc file. In addition to serving as the communication link between the two computers the disc file provides storage for both active and save programs. The remote control consoles are model 35 teletype machines.

A user uses the system in a manner previously described. He identifies himself and the programs he wishes to use. He can modify these programs and then get them run.

Inside the Datanet-30 are input-output buffered areas associated with each teletype station. These are operated in a flip-flop fashion so that input or output typing may continue in one part of the buffer while the other is connected to the disc unit. The executive program in the Datanet-30 is divided into two parts, a real-time part and a spare-time part. The spare-time part is mainly disc operations and certain teletype operations. The real-time part is entered via a clock-controlled interrupt 110 times per second in order to scan the teletype lines. As

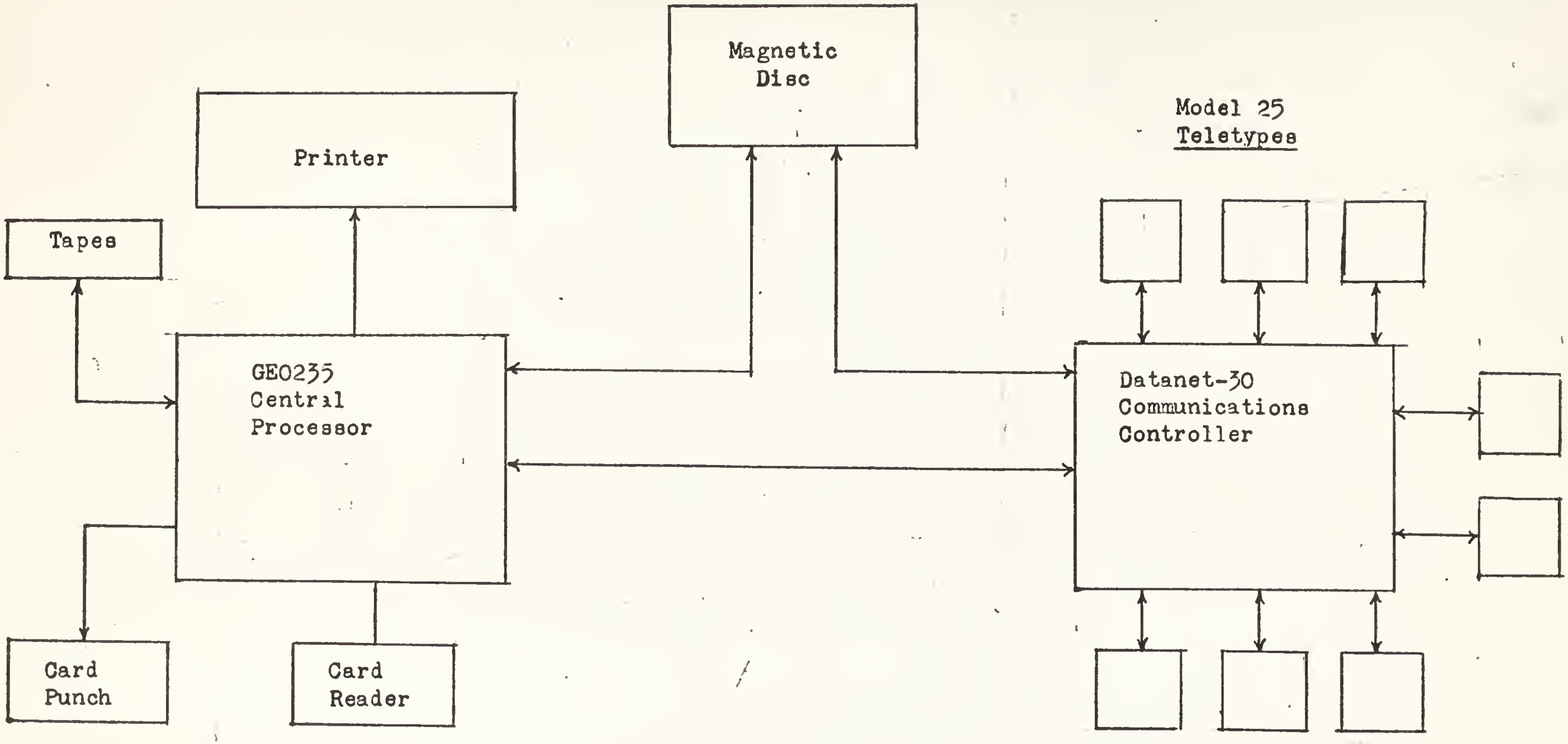


Figure 2.4.1. Schematic Configuration of Dartmouth Time-Sharing System Hardware

characters from remote stations are completed, the real-time part collects them into messages, and upon the proper command, interprets that message into various commands for the computer.

During part of the day the computation center uses the computer under monitor control and the time-sharing system is not compatible with this operation. In time-sharing operations, the user is restricted to a block of 6000 words, whereas, he has a larger block of storage under monitor operations.

The primary purpose of this system is to process small jobs, which supports the argument that time-sharing should be considered for smaller and more conventional installations as well as for major research. However, Dartmouth states that there can be fairly long waits of 5 to 10 seconds as the spare time tasks and run requests stack up. These stacks and delays are blamed on the disc file which is a relatively slow extension of core memory.

Dartmouth reports their time-sharing system is an extremely effective small job processor. The minimum amount of red tape required by the user to get on-line is purported to provide an accessibility equivalent to that of a desk calculator. Table look-up operations are reported to be made to order for this system.

2.5 Other Time-Sharing Systems

The previously mentioned time-sharing systems have special features which were significant to note. The systems mentioned cover major aspects of time-sharing systems but they do not cover all the time-sharing systems currently in use. Some other systems and their significant features are:

a. IBM Time-Sharing Systems

The QUIKTRAN time-sharing system is an experimental system designed by IBM. It is a one language system allowing up to 40 terminals on an IBM 7040/7044. The system was designed primarily for engineering and scientific problems in the desk calculator to small computer range. All man-machine communication is in the source language (a FORTRAN derivative) and is in a conversational mode. The translator has powerful debugging features but only with substantial degradation in executive efficiency (9, 10).

IBM also has an experimental, general purpose system which allows up to 20 users to communicate with a 7090 via 1050 console. It has a multi-queue scheduler with space-sharing and offers a few seconds response time. Whereas, the previous system is constrained to one language, this system can take FAP, FORTRAN, GPSS and PAT.

IBM also has another experimental system which uses a modified IBM 7044 with extended addressing capability and multiple-register facilities for pagination. This system has a large core capacity. It is designed specifically for experimental study of various time-sharing, space-sharing and multi-programming operations.

b. Stanford Time-Sharing System

The system uses a configuration of the PDP-1 and 7090 similar to that described in the SDC system. This system has 20 remote stations divided into 12 cathode-ray tube and eight teletype consoles. The swap time for user programs is 34 milliseconds for the PDP-1 and 600 milliseconds for the 7090. User programs are stored on an IBM 1301 disc. This system, used primarily as a teaching machine and for general computing, can take the following languages: MACRO, GOGOL and LISP on the PDP-1 and BALGOL, FORTRAN, FAP and LISP on 7090.

c. The Adams Associates-Key Data System

This system is an on-line packaged commercial data processing system with a PDP-6 central computer. The system can take FORTRAN and other engineering languages. The system originally had 16 leased-line teletype terminals but is being expanded to 256 terminals. It has dynamic core allocations with interpretive processing. Service is on a first-come, first-served basis with the quantum time either "to completion" or "to needed drum or disc access." The company states that response time should not exceed 250 milliseconds more than 10 percent of time.

d. Bolt, Beranek and Newman Hospital Computer System

The system uses a PDP-1 computer with a FASTRAND drum. It has an on-line MIDAS assembler plus a version of JOSS and can service 30 simultaneous teletypewriter users. The input queue is multi-level with a typical response time of 500 milliseconds. The executive program includes extensive common sub-routines to enable user programs to be prepared quickly.

2.6 Hardware Advancements in Time-Sharing Systems

In the previous discussions of time-sharing systems the emphasis has been on the software aspects. Creative computer personnel have taken the available equipment and, by ingenious programs, have tied them together into a time-sharing system. One of their major problems in developing a really sophisticated time-sharing system is the capacity of high-speed computer memory.

In response to this growing interest in time-sharing systems, computer manufacturers are designing new equipment which facilitates the operation of many remote stations. Control Data Corporation series 6600, General Electric series 600, and IBM 360 system, model 67, to name a few, have especially built-in features for time-sharing operations.

A representative example of some of this equipment is the IBM 360 system, model 67. It is a modular system which can have from one to four central processors with up to eight working core storage units of 256,000 bytes each; a byte is 8 bits, therefore 256,000 bytes of storage is equivalent to 42,666 storage words of 48 bit length. Under program control all processors can work with any or all memory units and other processors. The system is also partitionable so that a central processor, core storage and associated input-output equipment can function independently of the rest of the system.

In order to increase the high-speed memory, this system has a special relocatable memory feature which strikes at the problem raised by Schwartz (5). In time-sharing systems, programs most often have to be dumped out of core memory and provision must be

made for this dump. Since a program returned to core is unlikely to be assigned to the same physical area of core that it left, there must be a means of reassigning memory under program control so that instruction addresses can be assigned to new core locations.

Although the programmer or compiler assigns specific addresses to instructions, these are relative addresses and never refer to actual physical locations in the computer main memory. The computer's associated registers translate the relative addresses to physical addresses when the computer executes the instructions. This address changing is done within the computer and the programmer is unaware of the operation.

The IBM system has a 24-bit addressing scheme expandable to 32 bits. With a 24 bit address, the system behaves as if it had a core memory of 16 million bytes. With 32 bit address, this virtual storage expands to over 4 billion bytes. The 16 million byte memory divides into 16 segments, each of which is further subdivided into 256 pages. Each page contains 4,096 bytes. The time-sharing monitor allocates core to programs in units of 4,096 bytes and the units need not be contiguous. A special routine within the computer converts the logical address to the physical address in less than 200 nanoseconds.

To further stretch core memory, only one copy of commonly used assemblers, compilers and other routines are stored in core memory, and are made available to any program. To do this a reentrant technique of coding must be used. Compilation of one program can be interrupted, another program compiled, and the original compilation continued. This feature is particularly useful in

conversational-mode programming in which the computer handles a terminal instruction by instruction or, in case of problem-oriented languages, asks for a series of ordered data entries.

Another time-sharing problem brought out in the discussion of the Dartmouth System is the constraint placed on the system's speed by the data path via the external storage. The high speed data paths between central processors, core, auxiliary storage and input-output devices are critical to the system's efficiency. In the IBM system,channel controls have a major role in managing these data paths. The channel controllers improve system performance in three ways:

a. They provide more data paths so that any combination of central processor and memory can communicate with any input-output device. If a system has more than one channel controller it can bypass any inoperative data link and keep running.

b. They permit asynchronous operation of different parts of the system, so that each element can operate at its own optimum speed most of the time.

c. Because the channel controllers have direct access to main core, they free the central processor during input-output and data transfer operations. The system monitor tells the channel controller where it wants data put in or taken out and the channel controller takes care of the job without interfering with the central processor operation.

The system is designed so that the interconnections between the central processors, channel controllers and main memories are switched by a network known as distributed crossbar. This arrangement eliminates a central switching network that might cause total

system shutdown in case of malfunction. With distributed switching, a failure of one part of the switching network only disables the device associated with the switch, and the rest of the system remains operative.

To the remote user, this system looks and behaves as though all data and programs were in one big storage area. The whole hierarchy of storage is on-line and under program control: working core, high-speed drum, large capacity drum, high-speed disc, large capacity disc file and finally with data cell drives with bulk storage on magnetic strips. Thus, providing that sufficient storage devices are in the system, anything addressed in the 16 million or 4 billion byte virtual memory can be called in the CPU in about a second. Memory cost is, roughly, inversely proportional to access time.

3.0 Investigation and Results of Varying Some Basic Parameters of a Time-Sharing System

Computer simulation provides economically data for analyzing the interaction of the basic design parameters of time-sharing systems. Through simulations a good approximation of the actual performance of an operating system can be obtained. Some characteristics of particular importance in analyzing time-sharing systems are the average system response time, average number in queue awaiting service, average time a user is in the queue and the computational efficiency results of the system.

The system simulator chosen for gathering the desired data was one developed by Hatch (12) and Wilder (13) with several modifications made by the authors. The system and modifications are explained in detail in Appendices I - IV. This chapter will present and analyze the results of the actual computer runs resulting from varying several basic system parameters. As is often the case with simulations, these results suggest more simulation runs of interest and possible investigation.

3.1 A Time-Sharing System and Its Basic Inputs

The computer simulator used to generate data for this study is described in detail in Appendices I - IV. The computer used was the CDC 1604 located in the Computer Facility, U. S. Naval Postgraduate School. However, the simulation program assumes a computer with a three microsecond cycle time. Transfer times were computed on this basis.

In brief, the system assumes a configuration like Figure 2.1.2 and works as follows. As users' programs come into the system they are loaded onto a disc file which acts as a buffer for the central processor. The number of programs which can be loaded onto the disc is a variable which can be changed, and is dependent upon the storage capacity of the disc file. The disc capacity determines the number of stations allowed in the queue. An exchanger routine loads the user's program onto the drum file from the disc. In turn the scheduler routine controls the loading of programs into main memory. The system does not permit concurrent transfer of programs. An average access time of 15 milliseconds was used for all drum read and write transfer operations with a word transfer rate of 2.75 microseconds. For the disc file an access time of 140 milliseconds and a word transfer rate of 14 microseconds was used. As mentioned earlier a computer cycle time of 3 microseconds was assumed. Relocatability of programs was assumed and a memory allocation of 2 K words was provided for the resident portion of the time-sharing executive program. Individual program size is limited to 30 K words.

Each simulation run was made with two groups of input data. One set of data is a typical job mix which could be expected in

a large complex time-sharing system. For reference purposes this job mix is called the "Large Job" data. The other set of data is based on empirical data of program sizes and their frequency of occurrence collected at the Computer Facility of the U. S. Naval Postgraduate School. The job mix of this latter set, called "School Data" for reference purposes, consists of many small jobs with a small percentage of jobs of 2 K and 4 K word size. A mean arrival time of 300 seconds was assumed for each set of data. The other job characteristics are shown in Figure 3.1.1. The School Data could be representative of the types of jobs handled by the JOSS and Dartmouth time-sharing system.

By using either of the above sets of input data the characteristics of a job are created for each station by the program job generator subroutine called SET. A particular job type is determined by using the Monte Carlo technique of generating a uniform random number for the job probability. The type of job gives the basic input data for a particular station. Each job generated at a station is characterized by six variables. The time interval between arrival times, the first parameter, is assumed to be exponentially distributed. This assumption is based on queueing theory concepts and actual observations at System Development Corporation (12). As noted earlier the mean arrival time was assumed to be 300 seconds for both groups of data. The arrival time of each job was determined by adding the generated arrival time to the clock time. Load time, the second parameter, represents the time required to transfer the binary programs from the disc file to the drum file. Load time is a function of the

JOB TYPE	ACTIVE TIME	I/O TIME	REPEATS	MEAN SIZE	JOB PROBABILITY
1	3.0	1.0	9.0	2000	0.150
2	1.0	3.0	15.0	4000	0.200
3	3.0	1.0	9.0	6000	0.200
4	2.0	2.0	30.0	8000	0.150
5	2.0	1.0	30.0	12000	0.100
6	1.0	2.0	15.0	16000	0.050
7	2.0	1.0	12.0	20000	0.050
8	2.0	2.0	15.0	24000	0.050
9	2.0	1.0	15.0	28000	0.020
10	1.0	1.0	15.0	32000	0.030

LARGE JOB INPUT DATA TABLE

1	2.0	1.0	8.0	250	0.60
2	3.0	2.0	12.0	750	0.25
3	2.0	1.0	20.0	2000	0.10
4	3.0	1.0	30.0	4000	0.05

SCHOOL INPUT DATA TABLE

Figure 3.1.1

number of disc file accesses required to retrieve the full program and the disc file word transfer rate, times the size of program. The next three parameters, Active time, I/O time and Repeats, define the actual program operating characteristics. The sixth and last parameter, size (number of instructions), completes the job description. The last five parameters are determined by using a Gaussian Random Number generator with the mean values for each characteristic received as input from the job type. As soon as a job is completed, that is the number of repeats is zero, a new job is generated for that station (12). A sample of the job data generated by the modified simulator program for the school data is shown in Figure 3.1.2.

One of the most important aspects of a time-sharing system is computational efficiency which is dependent upon both hardware and software features of the system. Unfortunately, there is no one single definition of the term which is universally accepted. One way to define it would be: the total time the computer is performing useful computation on users' programs divided by the total time the computer is in operation. The total time the computer is in operation is the sum of idle time, overhead time, active service time and swap time. For example, in a four-hour period, what percentage of time is the computer actually servicing the users' programs? Hatch and Wilder defined efficiency as the average amount of time spent computing per response cycle, divided by the actual cycle time (12, 13). Since the latter definition is more widely recognized, it is used in this paper. For comparison purposes, statistics for both methods were gathered and, in general, the efficiencies for the former definition were found to be somewhat higher than those for the latter.

FIGURE 3.1.2

GENERATED JOB DATA

JOB NUMBER	CLOCK TIME	STATION NUMBER	JOB TYPE	ARRIVAL TIME	LOAD TIME	ACTIVE TIME	I/O TIME	REPEATS	SIZE
1	0	1	1	69.9181	.1546	1.8945	1.6466	8.6726	279
2	0	2	1	404.3657	.1553	2.5770	.9491	8.8736	231
3	0	3	1	579.6820	.1623	1.9300	1.1729	5.8906	254
4	0	4	1	334.6186	.1559	1.7070	1.1970	9.6339	232
5	0	5	2	16.3178	.1526	2.8179	1.8796	12.1617	739
6	0	6	1	83.9177	.1488	1.6943	.9131	10.1456	322
7	0	7	1	260.1744	.1454	2.0416	.8863	8.3893	247
8	0	8	1	643.2715	.1625	2.3026	.6961	6.5168	217
9	0	9	2	169.5604	.1831	2.7686	2.3709	13.3554	628
10	0	10	1	335.0660	.1128	1.5512	1.2414	8.1810	239
11	0	11	2	1185.8310	.1515	2.1834	1.8042	10.2970	608
12	0	12	1	234.7778	.1417	1.7837	1.4113	7.5893	243
13	0	13	1	117.2948	.1652	1.7330	.9556	8.5999	276
14	0	14	1	287.6530	.1380	2.2409	1.0532	9.2436	284
15	0	15	1	423.9947	.1542	2.3747	.7882	6.1708	204
16	0	16	2	584.0034	.1444	2.0338	1.9407	12.0940	702
17	0	17	2	473.3052	.1636	3.2207	2.1453	7.5287	819
18	0	18	1	295.0545	.1720	1.7823	1.2282	9.0985	278
19	0	19	1	564.4371	.0701	1.8382	1.0056	5.2124	334
20	0	20	3	37.8802	.1819	1.8858	.9356	23.5555	1340
21	169.2	5	1	436.3731	.1487	1.7986	1.1272	7.8735	271
22	184.0	1	1	1387.7225	.1438	2.0044	.9779	9.4975	217
23	204.0	6	1	312.5196	.1506	2.1075	.8347	9.1927	212
24	216.6	13	2	290.9017	.1496	3.7692	2.0169	10.3164	777
25	236.6	20	1	426.6998	.1302	1.9313	1.0297	9.8348	256
26	295.1	12	3	789.0401	.1581	2.2178	.9982	18.3916	1828
27	395.9	9	1	941.5680	.1569	2.1219	.8188	10.4382	191
28	406.9	7	1	1453.3206	.1412	1.6117	1.0580	7.7733	284
29	504.0	18	3	693.9764	.1751	2.0685	1.0385	21.1025	1748
30	550.1	10	3	620.4695	.1385	1.8824	1.0319	21.0849	2376

FIGURE 3.1.2 (CONTINUED)

31	556.0	14	1	566.0410	.1109	1.9403	.6744	6.3813	287
32	595.1	4	3	853.9918	.1938	1.9436	1.0691	16.4700	2045
33	601.6	6	3	928.7797	.1381	1.5345	1.1944	21.1199	2186
34	717.9	5	2	993.2194	.1848	2.6183	2.9028	10.2832	988
35	720.8	15	1	1260.2161	.1580	1.6557	.8439	9.8704	284
36	773.4	20	1	1013.1358	.1318	2.0973	.8541	9.3055	308
37	789.6	19	1	869.0159	.1605	1.8761	.9416	10.8719	247
38	833.4	2	1	1258.4865	.1056	2.3686	.7956	5.9134	164
39	834.2	3	1	884.4019	.1403	1.9744	.9276	7.3918	244
40	838.3	14	2	1668.3680	.1946	3.6357	2.0493	11.8364	562
41	874.3	13	3	1701.5232	.1732	1.2140	.9063	14.0420	1842
42	912.1	17	1	1284.1590	.1252	2.5046	.9760	8.1150	238
43	932.4	8	2	1032.9096	.1690	2.3899	1.7410	11.8101	1018
44	1026.7	16	1	1071.0495	.1579	1.5652	1.0917	6.7148	205
45	1198.6	3	1	1236.9088	.1048	2.1586	1.1829	8.4766	237
46	1230.2	19	2	1536.0248	.1636	3.4336	2.3864	12.1512	567
47	1354.1	16	1	1388.8126	.1648	1.8948	.6633	8.0645	246
48	1397.8	10	3	1910.1694	.1585	2.4741	.7957	26.0666	2100
49	1482.3	9	2	1495.8972	.1069	2.9709	2.2674	10.6480	832
50	1499.8	20	3	1508.9321	.1841	2.1791	.6567	17.1783	2720
51	1591.9	18	1	1848.0748	.1399	2.1218	1.0134	9.1323	206
52	1622.3	4	1	1985.3751	.1378	2.0592	.9366	11.7097	228
53	1663.3	12	1	2989.0393	.1737	2.5383	.8947	9.0093	284
54	1787.3	2	2	1999.7365	.1618	3.3588	2.1982	10.0137	704
55	1809.6	15	1	2301.7210	.1300	2.1695	.9575	8.1808	240
56	1811.0	6	1	2074.8220	.1589	1.4057	.8813	7.1032	194
57	1833.6	5	2	1854.4311	.1301	1.9469	1.8892	13.2368	673
58	1854.7	8	1	2124.8013	.1382	2.0881	.8620	6.7687	239
59	1891.2	3	2	1894.0855	.1244	3.5172	1.9243	11.0228	885
60	1895.8	7	3	2011.2200	.1709	2.0597	.9249	20.2755	1693
61	1913.6	11	1	2382.5608	.1823	2.7367	.8054	6.8301	274
62	1919.7	16	3	2827.3833	.1440	1.9682	.8639	18.4851	2229
63	1954.6	17	2	2152.4115	.1035	3.0160	1.7287	11.3402	726
64	2021.7	1	2	2196.4382	.1559	2.2635	2.1896	14.0189	784
65	2168.4	13	3	2340.4893	.1463	1.8980	.9954	15.1789	2222
66	2395.0	18	2	2743.1894	.1183	2.9611	2.1367	9.4179	818
67	2493.4	9	1	2787.7993	.1388	2.1606	1.0464	8.4235	237
68	2505.5	6	1	3311.7583	.1591	2.5373	.8910	8.1543	246
69	2534.6	20	1	3025.3053	.1496	1.7418	.9163	8.3617	250
70	2659.6	8	2	2821.9794	.1574	3.4937	1.9050	14.0107	906

User requests were handled on a single "round-robin" basis with a quantum determination for each response cycle. It was decided that the simulation data should be collected from a time-sharing system under a "worst-case" circumstance. The "worst-case" for a time-sharing system is when the system is in a steady state with a fairly stable queue. In order to find this steady state the same program with the same input data was run for one, four and eight hours simulations. It was found that a four hour run provided the steady state condition and therefore all results are based on program runs of this duration.

Another important characteristic of a time-sharing system is the total time users actually spend in the system before they are completely serviced. A subroutine to accumulate these times and display them in histogram form was added to the original program.

In order to compare the effects of computer center policies, three basic schemes of time-sharing operations were investigated.

a. A normal run with quantum time re-determined if the users finishes early.

b. A normal run without re-determining the quantum when a user finishes early.

c. A normal run with quantum time re-determined and with production programs, called background users, receiving a quantum of time each response cycle only if the queue is not full.

3.2 Investigation of Some Methods of Determining the Quantum Time

One of the most important parameters of a time-sharing system is the quantum of time permitted each user. The quantum time given each user is a compromise between two opposing desires (1) to process the user's program as quickly as possible so as to reduce his time awaiting complete servicing; and (2) to reduce the response cycle time for the system. The resulting compromise directly affects the computational efficiency. The ideal system would be to have the response time approach zero, total user service time approach zero, and computational efficiency approach 100%. This ideal system is not possible but can be approached by the determination of a quantum which optimizes these three desires.

In order to investigate the affects of the method of determining the quantum time, four methods were investigated.

1. Q_1 = FNCYCTM/FNQUE with second phase determination
2. Q_2 = (FNCYCTM * (TCYTM - CSOVRHD)/TCYTM)/FNQUE + (TEDUMP - TELOAD)/FTCYCNT
3. Q_3 = FNCYCTM/FNQUE without second phase determination
4. Q_4 = 200 milliseconds, a fixed quantum

FNCYCTM = the desired response time, set at two seconds for this investigation

FNQUE = the number of users in the queue at the start of the "round-robin" cycle

TCYTM = cumulative cycle time, in seconds

CSOVRHD = cumulative overhead

TEDUMP = cumulative time to dump programs

TELOAD = cumulative time to load programs

FTCYCNT = total number of quanta given at current point in program.

Quantum determination methods Q_3 and Q_4 are self-explanatory but Q_1 and Q_2 need further explanation. In order to facilitate the setting of the quantum time clock, Q_1 has a procedure, called second phase quantum determination, whereby the computed quantum time is rounded to the next lowest value of a set of quantum times. For this program the values used are 200, 150, 100, 75, 50 or 25 milliseconds. Q_3 makes no provision for facilitating the setting of the quantum clock. Quantum determination methods Q_1, Q_3 and Q_4 do not consider overhead time in their computation, whereas Q_2, which is a version of the method used at SDC, attempts to keep to the desired response time of two seconds by considering the portions of cycle time used for overhead and swap operations.

Each method of quantum determination was run for the two sets of data, i.e., the Large Job Data and School Data, for the three policies of computer center operations previously described. The runs were made for a maximum number of 20 user stations. Figures 2.1.1 through 2.1.8 (these figures and all other figures referred to hereafter are in Appendix V) summarize the results of these runs.

Comparisons of the results for the two sets of data show that the smaller-job mix which makes up the School Data can be handled by every quantum determination method more efficiently. The difference in computational efficiencies for the two sets of data range from approximately sixteen to twenty-nine percent for early termination allowed and background users allowed methods. The

same approximate ranges held true for early termination not allowed, except for Q_2 quantum determination method. The difference between the data in this case is almost 39%. The reason for this can be seen from Figures 3.2.9 and 3.2.10 which show histograms of the total time in the system and distribution of total time. The swap time for the large job mix is about 9 times the swap time for the smaller job size data. The overhead for the large job data is also greater.

For the School Data, there is little significant difference between the three computer center policies of recomputing the quantum after early termination, not recomputing the quantum and allowing background users except for the Q_3 method. Figure 3.2.3 shows that in this case the policy of not redetermining the quantum after early termination gives a significantly higher computational efficiency. The average response time is somewhat greater, about 270-310 milliseconds, but the average number in the queue is less. Because of the higher computational efficiency the number of stations (users) completely serviced is greater. The reason for this is primarily the difference in overhead requirements for the policies. The overhead time for quantum redetermination is about 78% of the other two. Also, the swap time for this program is about 75% of the other two.

As far as computational efficiency is concerned, the Q_2 method gives the lowest efficiency of any of the quantum determination methods for both job mixes. For the Large Job data mix, the computational efficiency is a very low 20%, whereas the next lowest is the Q_1 method which averages about 48%. While the computational efficiency for the School Data for Q_2 is still less than Q_1 the

range is not as great. This is caused primarily by the average quantum time for Q_1 being four times greater for Q_2 for the large job mix whereas, for the School Data, Q_1 is only two and one-half times as great.

It should be noted that the highest efficiencies are obtained by fixing the quantum for each user at 200 milliseconds, which is the Q_4 method. The efficiencies with this method are somewhat higher than the Q_3 method, particularly for the Large Job mix data. A comparison of the overhead times for the School Data job mix shows that, for the quantum redetermination policy, the overhead time for Q_4 is 15% less than that of Q_1 and 48% less than Q_2. However, the average response time for Q_4 is greater than Q_1 and Q_2 but less than Q_3. The average number in the queue for Q_4 is less than the other three methods because the average quantum time is greater for each policy for both groups of data.

Because the number of simulations for investigating other parameters of a time-sharing system is a permutation of the number of ways the quantum could be determined, the two sets of data and the other parameters, it was decided that further investigations should be made using only one method of quantum determination. Method Q_1 was chosen because it offered the best compromise between computational efficiency, average cycle time, average number in queue, average quantum time, and number of stations serviced.

Some of the effects of the four methods of quantum determination are illustrated in the form of (1) histograms showing the total time in the system and (2) system utilization diagrams showing the percentages of idle, overhead, active user time and swap time. See Figures 3.2.11 through 3.2.18. Histograms are provided only

for the computer center policy of redetermining the quantum for early terminations since they are representative of all methods.

3.3 Effects of Varying Response Cycle Time

Twenty-five four-hour simulations were run varying the response cycle time from 2 to 10 seconds in increments of 2 seconds. The variable quanta allowed ranged from 200 to 25 milliseconds depending on the number of stations in the queue. The Large Job data were used to obtain job parameters.

The study allowed a maximum queue of 20 stations with 20 and 30 users, a maximum queue of 30 with 30 and 35 users, and a maximum queue of 35 with 35 users.

Figures 3.3.1 to 3.3.5 give the results of these simulations.

It is evident that as the response cycle time increases the computational efficiency also increases and the average service time improves.

Analysis of Figure 3.3.1 shows that the maximum computational efficiency is reached when the response cycle time is set at 6 seconds. This indicates that the maximum computational efficiency obtainable for this set of conditions and for the variable quanta range being used has been reached. Figure 3.3.6 was obtained by using a variable quanta range of 400 to 100 milliseconds. It is seen that the computational efficiency was further improved and that a different variable quanta range is probably needed for a response cycle time of 10 seconds.

The computational efficiency of a system using the variable quanta range of 200 to 25 milliseconds and the standard method of computing the quantum used in this thesis appears to have an upper bound of approximately 63%.

3.4 Investigation of Varying the Number of Remote Stations in the System

Also of interest is the affect that the number of stations serviced by the system has upon the operating characteristics. Data was collected for the two sets of data for quantum determination methods Q_1 and Q_4 (described in Section 3.2). A two second response cycle was assumed and the maximum number of users permitted in the system at any time was set to twenty.

Data was accumulated for runs using the three computer center policies (see Section 3.1) for ten, twenty, thirty, forty and fifty remote stations.

A comparison of the results for the two groups of data, Figures 3.4.1 - 3.4.16 (data for twenty stations are given in Figures 3.2.1, 3.2.4, 3.2.5 and 3.2.8) show that the computational efficiency for the School Data job mix is significantly higher than the Large Data job mix. The average response time is also significantly less for the School Data. The greatest difference in computational efficiency between these two job mixes appears when twenty stations are allowed and the smallest difference occurs for ten stations. The difference between job mixes for thirty, forty and fifty stations is approximately the same.

With the exception of the results for the School Data for quantum determination method Q_4 for ten and twenty stations (Figures 3.4.2 and 3.2.4) the computational efficiency is greatest for ten stations. The computational efficiency for thirty, forty and fifty stations are approximately the same. This suggests that the system becomes saturated after thirty stations and adding more stations has little effect. The average number in the queue bears this

statement out. With the exception noted for Q_4 the average cycle time increases as the number of stations increase. A significant jump in times for ten to twenty stations can be noted.

A comparison of the results for quantum determination method Q_1 and Q_4, with ten and twenty stations, for system policy of re-determination of quantum time allowed, will help to explain the exception noted for Q_4 with the School Data. The results of the simulation runs for the School Data are:

	QUANTUM DETERMINATION METHOD Q_1		QUANTUM DETERMINATION METHOD Q_4	
TIME ALLOWED	10 STATIONS	20 STATIONS	10 STATIONS	20 STATIONS
IDLE	2043 sec.	37 sec.	3364 sec.	66 sec.
OVERHEAD	2788	3945	748	3379
ACTIVE	9277	9869	10208	10479
SWAP	291	549	80	474
EFFICIENCY*	64%	68%	71%	78%

* Determined by dividing active service time by total simulation time (4 hours)

As can be noted from the above data the active service time for both Q_1 and Q_4 is higher for twenty stations. Computer efficiency is also higher for twenty stations. This is not the case for the large job data mix. The computational efficiency, for both methods Q_1 and Q_4, decreases as the number of stations increases. This relationship also holds for the School Data in going from twenty to fifty stations. This relationship can be seen in the histograms of the users total time in system and time allocations. Representative histograms of each run are shown for ten and thirty stations as

Figures 3.4.17 - 3.4.24 (the histograms for twenty stations are shown as Figures 3.2.11, 3.2.14, 3.2.15 and 3.2.18). The histograms for forty and fifty stations are almost carbon copies of the thirty stations' histograms.

The greatest computational efficiency is found at ten stations for the system policy which allows background users. The reason for this is that the idle time which is proportionately high for 10 stations, is used to process the background users job; whereas the computer is standing by idle for the other system policies. As the number of stations increases, this idle time decreases and becomes insignificantly small, therefore nothing is gained by permitting background users. As the number of stations increase the system characteristics for this policy and the policy of redetermining the quantum become the same. It is significant to note that the highest computational efficiency is for quantum determination method Q_4 but this method also gives the greatest response time. The average number in the queue is somewhat greater in all cases for this system policy.

For twenty stations, there is no significant difference in the computational efficiency for each system policy. The other system characteristics for twenty stations are discussed in Section 3.2. As noted above, the system becomes saturated after thirty stations and there is no significant difference in computational efficiency for each system policy after thirty stations. However, it should be noted that the computational efficiency for the system policy of not permitting quantum redetermination upon users terminating early is consistently higher for each run. This is due to the overhead being about ten percent greater for the quantum

redetermination method. The swap time for the redetermination method is also somewhat higher.

The computational efficiency using Q_1 is higher than Q_4 for both sets of data for ten stations whereas the reverse is true for twenty stations. Beyond the saturation point there are no significant differences in either method although it should be noted that for Q_4 the computational efficiency is slightly higher for the School Data whereas Q_1 is slightly higher for the Large Job data mix.

Above saturation point there is no significant difference in the average overhead for any of the runs. For twenty stations the average overhead for the School Data for Q_1 is less than for Q_4, although for the Large Job data the Q_1 average overhead is greater.

In general it can be said that as the number of stations increase, the computational efficiency decreases, the average cycle time increases, the average queue increases until the saturation point is reached and then these characteristics level out.

3.5 Effects of Varying Queue Size and Number of Stations

Sixteen four-hour runs were simulated to investigate the effect of varying the maximum queue size in the following manner 20, 25, 30, 35, 40, and 50. The number of stations were allowed to vary from 10 to 50 stations in increments of 10 for each maximum queue size. Since the results for 10 users is the same for all maximum queues from 20 to 50, only one run was necessary for 10 users. Similarly, runs could be reduced for other simulations. Figures 3.5.1 and 3.5.2 show the results of these simulations.

The job profiles for the Large Job data were used to determine job parameters. A response cycle time of 2 seconds was postulated and the quanta offered ranged from 200 milliseconds to 25 milliseconds. Figures 3.5.1 and 3.5.2 show that the computational efficiency was greatest with 10 users in the system and rapidly decreased as the number of users increased. Although computational efficiency was highest for 10 users, the system idle time was also greatest at 9.8% of the total time as compared with less than .2% for all other simulations.

The best level of operation to meet the above characteristics and maintain a high computational efficiency is a maximum queue size of 20 and maximum number of twenty stations. However, if the design conditions were to be relaxed slightly, that is, to allow a 4 second average cycle time and a lower computational efficiency, the system could be expanded to a maximum queue of 30 and a maximum of 30 stations. Analysis of Figures 3.5.1 and 3.5.2 shows an insignificant difference between the maximum queues of 20, 25, and 30 with a maximum of 30 stations. The increase in average response

cycle time between a queue of 20 and a queue of 30 is less than one second, however the average time to service a user to completion would be increased by 10 minutes. This increase is considered to be tolerable.

Figures 3.5.3 to 3.5.7 illustrate the time a user would be in the system for a maximum queue of 20 with 10, 20, and 30 stations and maximum queues of 25 and 30 with 30 stations.

Another set of simulations with the same conditions as above were run using the job profiles of the School Data to determine the job parameters.

Analysis of Figures 3.5.8 and 3.5.9 shows that as the jobs tend to become small in size, which is generally expected in an academic environment, the swap times are smaller and therefore computational efficiency increased. Further analysis revealed that because of the small job sizes the drum capacity was never exceeded and the system could handle 50 stations with a maximum queue of 50. Although the average cycle time is not significantly different between 10 stations and a maximum queue of 20, and 50 stations and a maximum queue of 50, there is a significant increase in the total time for a user to be serviced to completion in the system, from approximately 9 minutes to 61 minutes.

Again, if the desired goals are relaxed or changed the proposed system could be expanded to 30 stations with a maximum queue of 30. The effect of this decision would be a decrease in computational efficiency of 15% and an increase in the average service time of 12 minutes when compared with the results that could be obtained with a 20 station system and a maximum queue of 20. The next step to 40 stations and a maximum queue of 40 would result

in a decrease of computational efficiency of 25% and an increase in the average service time of 25 minutes when compared with the results of the 20 station system and a maximum queue of 20.

4.0 Conclusions and Recommendations

An investigation of the effects of varying four basic software parameters in time-sharing systems was conducted and presented in Section 3. The simulation results offered much data for analysis and also presented some new areas for further investigations.

When policies and goals are established for a time-sharing computer center consideration must be given to the many variables which are interdependent. Since the computer center will be torn between maximum utilization of equipment, computational efficiency and user satisfaction, trade-offs are necessary between these variables.

The simulations have shown that for each desired response cycle time there is a particular variable quantum range which tends to maximize the computational efficiency.

The effect of various job mixes was also evident from the runs utilizing simulation with large and small jobs. It is of utmost importance that valid data be utilized in simulating a time-sharing system if goals and equipment being considered are to accomplish the jobs to be done.

A review of all the results shows that the computational efficiency is greater, the average number in the queue is smaller, the average response time is less, and total time the user is in the system is less for small data job mixes, such as the School Data, than for the Large Job data mix. Overall, the computational efficiency for the School Data averaged about fifteen percent greater than for the Large Job data. This is primarily due to the swap time for the smaller job mix being much less than the larger job mix.

With the one exception noted (for Q_4 with the School Data and increasing number of stations from ten to twenty) the computational efficiency decreases as the number of stations increases until a maximum efficiency is reached for that data mix. When quantum determination method Q_1 is used the response time also increases with the number of stations. However, with a fixed quantum of 200 milliseconds per user (Q_4), the response time is considerably longer for ten stations, decreases from ten to twenty stations and then increases as the number of stations increase but never reaches the response times for ten stations. The length of time the user spends in the system also increases as the number of stations increases. The conclusions hold until the system saturation point is reached. At this point the trend of these characteristics level off, although another system characteristic, the number of users turned away, continues to increase as the number of stations increase.

When the three computer center policies are considered, the policy of permitting background users gives a significantly higher computational efficiency for ten stations in all runs. This is because the computer is always computing either on background users' programs or remote station users' programs when they request service. When the other two policies are in effect, the computer is idle when there are no remote station users. However, the policy of permitting background users loses its edge as the number of stations increase, because the idle time for the other systems becomes insignificantly small, being less than one percent of the total time. As the number of stations increases the policy of not redetermining the

quantum gives a slightly higher computational efficiency. The overhead required for this policy is somewhat less than that required for the other two policies.

One of the parameters of greatest importance in a time-sharing system is the method of determining the quantum time allowed per user. This determination vitally affects the computational efficiency, the average number in the queue, the response time and the total time a user spends in the system. If the quantum methods investigated were ranked with desirable system characteristics being first the following table would result. The table results from varying the quantum determination method and holding everything else constant.

RANKING	COMPUTATIONAL EFFICIENCY	AVERAGE NUMBER IN QUEUE	AVERAGE RESPONSE TIME	TOTAL TIME IN SYSTEM
1	Q_4	Q_4	Q_2	Q_4
2	Q_3	Q_1	Q_1	Q_3
3	Q_1	Q_3	Q_3	Q_1
4	Q_2	Q_2	Q_4	Q_2

On this evidence, it appears that Q_4, i.e., a fixed quantum time, gives the best compromise for a time-sharing system. However, this finding is inconclusive and the quantum should be further explored by varing the maximum quantum permitted a user for the Q_1, Q_2 and Q_4 methods. Quantum method Q_3 varies only with the number in the system and the desired response time; little would be gained by continuing investigation of this method. The other methods should be investigated over a quantum range of one to two seconds as a maximum and 25 to 200 milliseconds as a minimum.

The results of the investigation of these ranges will permit a better decision on the optimum quantum determination method.

A response time should be established which is in consonance with the type of job mix to be run. If the job mix is such that little human interaction is necessary, a short response cycle is probably desired. If, however, the jobs are of a type which require a large amount of on-line programming and debugging, the response time could be made longer to take advantage of the rather slow human reaction time while the user is communicating via typewriter with the computer.

BIBLIOGRAPHY

1. Corbato, F. J., M. M. Daggett, R. C. Daley, R. J. Creasy, J. D. Hellwig, R. H. Orenstein and L. K. Korn, The Compatible Time-Sharing System - A Programmer's Guide, The M.I.T. Press, Cambridge, Massachusetts, 1963.

2. Gallenson, L. and C. Weissman, "Time-Sharing Systems: Real and Ideal," SDC Document SP-1872, 11 March 1965.

3. Samuel, A. L., "Time-Sharing on a Computer," New Scientist, Vol. 26, No. 445, 583-587, 1965.

4. Coffman, E. G., and B. Krishnamoorthi, "Preliminary Analyses of Time-Shared Computer Operation," SDC Document SP-1719A, August 14, 1964.

5. Schwartz, J. I., E. G. Coffman, and C. Weissman, "Potentials of a Large Scale Time-Sharing System," SDC Document SP-1723, July 28, 1964.

6. Corbato, F. J., M. Merwin-Daggett and R. C. Daley, "An Experimental Time-Sharing System," Proceedings - Spring Joint Computer Conference, 1962. Vol. 21, The National Press, Palo Alto, California. 335-343, 1962.

7. Shaw, J. C., "The JOSS System," Datamation, 32-36, 1964.

8. "The Dartmouth Time-Sharing System - A Brief Description," Dartmouth College Computation Center Paper, October 19, 1964.

9. Dunn, T. M. and J. H. Morrissey, "Remote Computing - An Experimental System - Part 2: External Specifications," Proceedings Spring Joint Computer Conference, 1965, Vol. 25. Spartan Book, Inc., Baltimore, Md, 413-423, 1964.

10. Keller, J. M., E. C. Strum and G. H. Yang, "Remote Computing - An Experimental System - Part II: Internal Design," Proceedings Joint Computer Conference 1964, Vol. 25, Spartan Book, Inc., Baltimore, Md., 425-443, 1964.

11. "A New System for Time-Sharing," IBM Computing Report for the Scientist and Engineer, Vol. 1, No. 1, 8-9, 1965.

12. Hatch, R. R, An Investigation of Program Exchange Methods for A Multi-Programming Environment, U. S. Naval Postgraduate School M.S. Thesis, 1964.

13. Wilder, W. G., An Investigation of the Scheduling Aspects of Multi-Programming, U. S. Naval Postgraduate School M.S. Thesis, 1964.

14. Corbato, F. J., A. Perlis, J. I. Schwartz, J. McCarthy and R. Patrick, "A Panel Discussion on Time-Sharing," Datamation, 38-43, 1964.

15. Critchlow, A. J. "Generalized Multiprocessing and Multiprogramming Systems," *Proceedings - Fall Joint Computer Conference, 1963*. Spartan Books, 1963.

16. Fine, G. H. "Preliminary Investigations in Time-Sharing Simulation," SDC Document TM-2203, 15 January 1965.

17. Fine, G. H. and P. V. McIsaac. "Simulation of A Time-Sharing System," SDC Document SP-1909, 29 December 1964.

18. Flores, Ivan. "Derivation of A Waiting-Time Factor for A Multiple-Bank Memory," *Journal of the Association for Computing Machinery*, Vol. 11, No. 3, pp. 265-282.

19. Glaser, E. L. and F. J. Corbato. "Introduction to Time-Sharing," *Datamation*, 24-27, 1964.

20. Harris, R. P. "The PDP-6-Time Sharing Hardware," *Datamation*, 51-54, 1964.

21. Kinslow, H. A. "The Time-Sharing Monitor System," *Proceedings of Joint Computer Conference Fall 1964*, Vol. 26, Spartan Books, Inc., Baltimore, Md., 443-454, 1964.

22. Krishnamoorthi, B. and R. C. Wood. "Time Shared Computer Operations with Both Interarrival and Service Times Exponential," SDC Document SP-1848/000/00, October 30, 1964.

23. *Massachusetts Institute of Technology Project MAC - Progress to July 1964*, MIT, Cambridge, Mass., 1964.

24. Schwartz, J. I., E. G. Coffman and C. Weissman, "A General Time-Sharing System," *Proceedings Spring Joint Computer Conference*, 1964.

25. Schwartz, J. I., "The SDC Time-Sharing System-Facilities, Services and Potential," *Datamation*, 28-31, 1964.

26. Schwartz, J. I. "The SDC Time-Sharing - Service Routines and Applications," *Datamation*, 51-55, 1964.

27. Shaw, J. C., "JOSS: A Designer's View of An Experimental DN-T Line Computing System," *Proceedings Fall Joint Computer Conference 1964*, Vol. 26, Spartan Books, Inc., Baltimore, Md., 455-463, 1964.

APPENDIX I

PROGRAM SIM

Program SIM is a simulation program written by Lt. W. G. Wilder and Lt. R. R. Hatch as part of their theses (12, 13) studying two aspects of time-sharing systems. The simulator program uses a Monte Carlo sampling technique for estimating results of an actual time-sharing configuration.

The hardware characteristics used in the simulator were patterned after the AN/FSG-32 (Figure 2.1.3 on page 10) configuration at the System Development Corporation, Santa Monica, California. The authors have assumed that these parameters, as used by Hatch and Wilder, were reasonable.

The general assumptions of the model are:

1. The system employs a scheduled or "round-robin" queue in which the computer makes a list of the channels with a request for service. During the response cycle the computer serves only these requests, giving each user a quantum of computer time. Also, during the response cycle other channels may have generated requests. After the conclusion of the cycle, the computer repeats the process, listing all the channels with a request for service, including those users not serviced to completion on the first cycle, and gives each user a quantum of computer time.

2. Computation and swapping of programs can not be overlapped.

3. Only one new user can be introduced into the system during a response cycle and this process can not be overlapped with other computer operations.

4. The required I/O equipment is available at all times. Differences in I/O such as tape transfers, searches, or outputs

to reactive devices are not recognized by SIM.

5. Once a program is loaded into the system it has an active period followed by an I/O period, during which no service is required from the central processor. This cycle is repeated until the job is completed or there are no further repeats required.

The authors have made several changes and additions to the simulator program which corrected deficiencies and slight programming errors in the original version of Program SIM.

The major changes incorporated were:

1. Alteration of the formula for calculating the job arrival times. In accordance with texts on queuing theory it is more proper to simulate a negative exponential arrival rate by the ln (1/1-y), where y is a uniform random number between 0 and 1.

2. Computation of the quantum during the first phase was changed from fixed-point to floating-point arithmetic (2 statements beyond statement 1530 in Program SIM). This change made a significant difference in the quantum offered when the number in the queue exceeded the response cycle time (in seconds) desired.

3. A different random number generator was used because the authors had doubts about the validity of the random number generator in the original version of SIM. The new random number generator was supplied to the authors by Professor R.H. Shudde of the Operations Analysis Department.

4. The program was translated to run in the FORTRAN 63 language for the CDC 1604 computer. This enabled the simulations to be run faster after the initial compilation and also allowed a more efficient random number generator written in CODAP to be used.

The four-hour simulation runs required 4 to 6 minutes per run on the CDC 1604.

5. Two subroutines, ACCUM and HIST, were added to allow the accumulation of frequencies for various data and to plot a histogram utilizing subroutine DRAW from the Computer Facility's library.

6. The method of determining the load time for each job was programmed to have some correlation with the size of the job. The load time takes into account the time to transfer the job and the access times to physically locate the job on the storage medium.

Three minor changes worthy of mention are:

1. A portion of the program that gathered statistics, immediately after the quantum was determined, were relocated due to the failure to gather the statistics at the proper time.

2. The portion of the program which redetermined the quantum after a user terminated service early, jumped to a statement which inadvertently dropped the last user from the queue. This was corrected by jumping one statement beyond the statement originally jumped to in the program.

3. "Stations Serviced" was defined to be stations which have completed service and does not include those still in the system.

Program SIM allows the user to investigate both hardware and software characteristics of proposed system designs. In this thesis the authors were primarily concerned with investigating in detail the scheduling aspect of time-sharing within a fixed hardware configuration.

Program SIM was modified so that input data could be read in with two read statements (Statements 8 and 14). Statement 8 reads in the job profile or characteristics, described in detail in 3.1.1 (Figure 3.1.1 on page 30). Since the array, AVERAGE, is of order 10x10, ten data cards must be read, however the job profiles may vary depending upon the empirical data to be used.

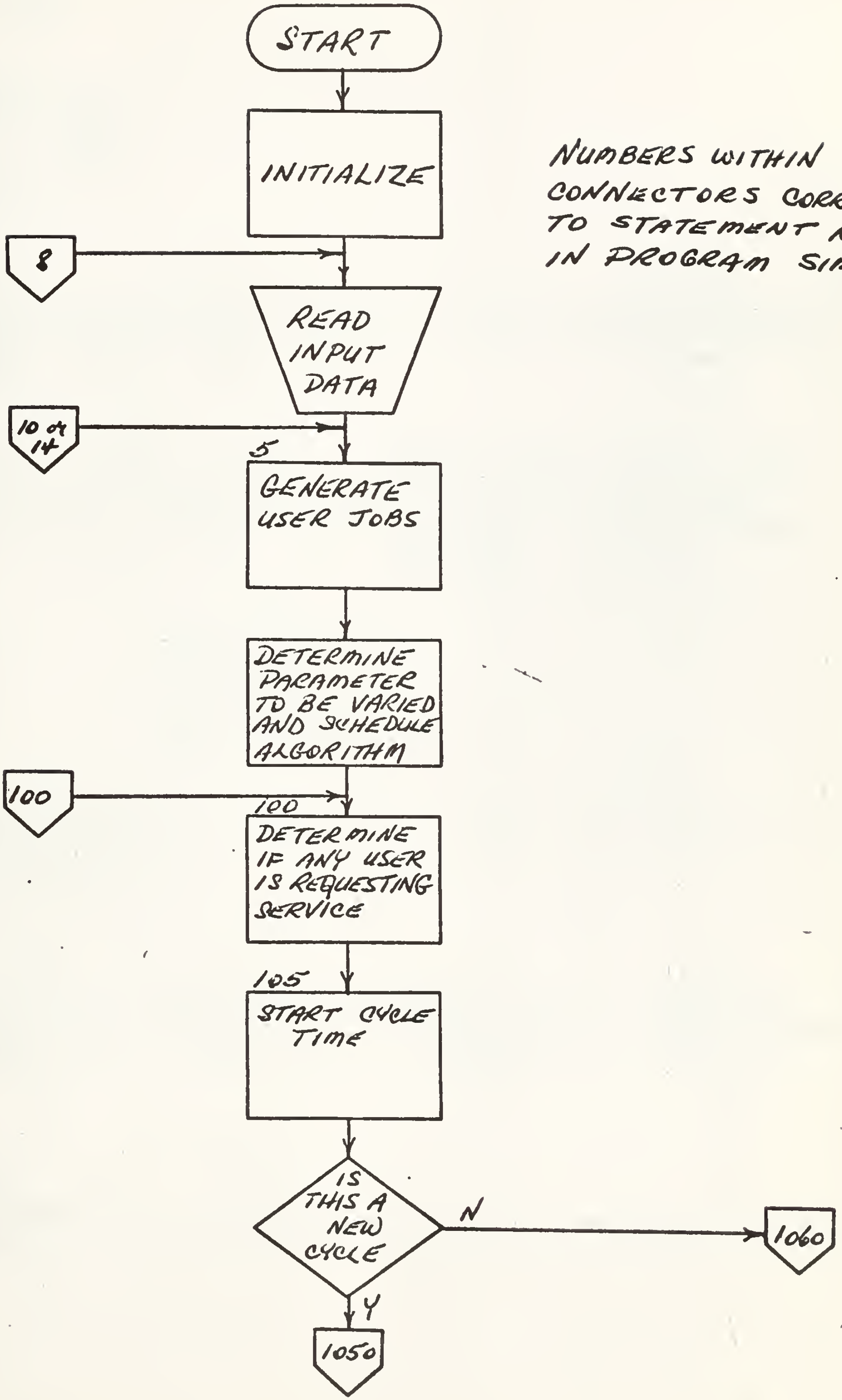
Statement 14 reads in the variable parameters NCYCTM, NU, and IQMAX, the index, ISKEDTP, which designates the scheduler algorithm to be used, the indices, ISTOPF and IFINIS, which control reading of additional data cards, the index, IVARY, which designates the parameter to be varied, and the index, IOUTCON, which controls print-out of the jobs generated.

The program could be further modified to read in the variable quanta array QA, the scheduler overheads SOVRHD1, SOVRHD2, and SOVRHD3, and MAXCOR, MAXDRM, and MAXDSC rather than having them fixed in the program.

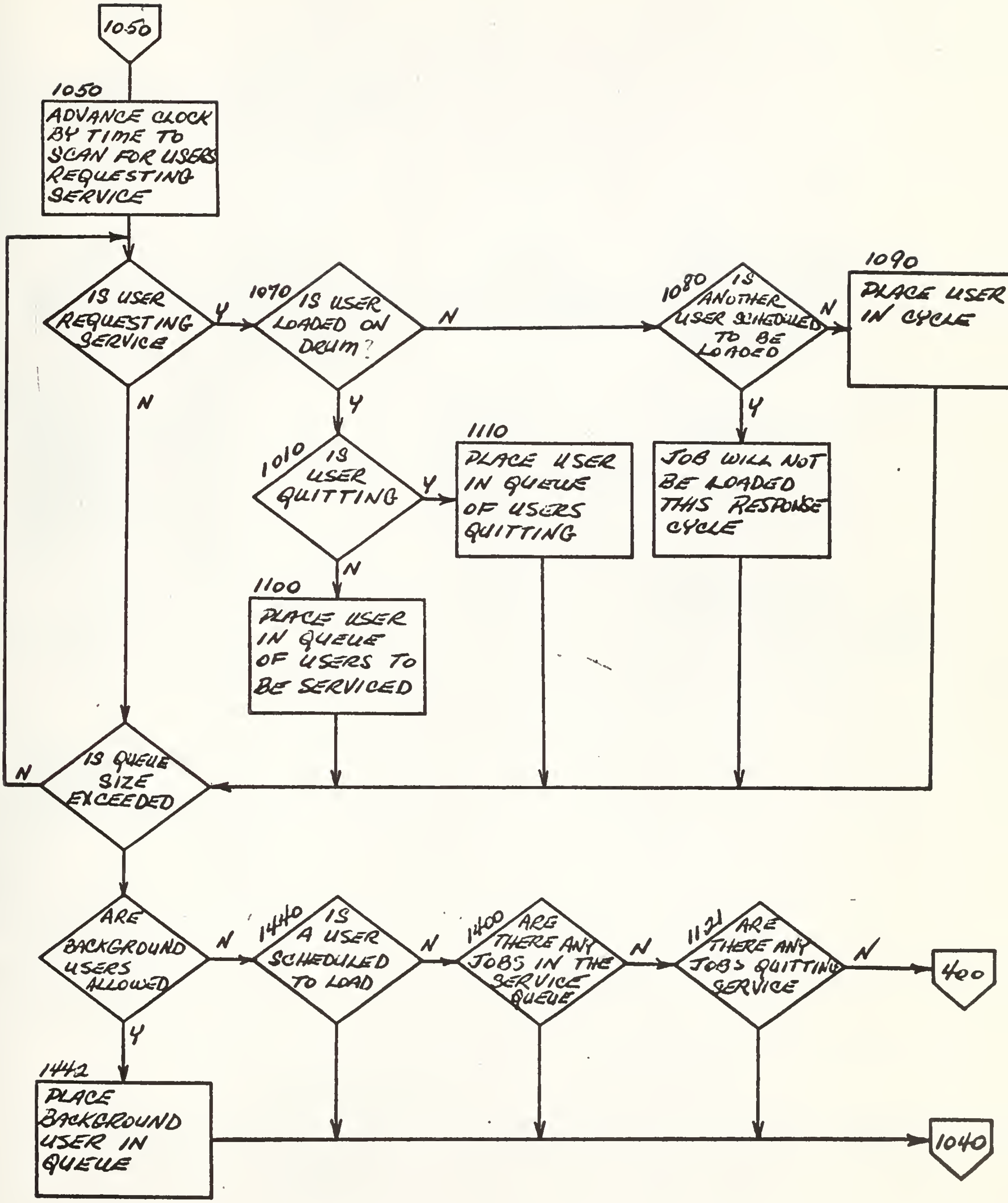
Greater details about the simulator program are contained in the flow diagram in Appendix II, a copy of the program in Appendix III, and a glossary of FORTRAN names in Appendix IV.

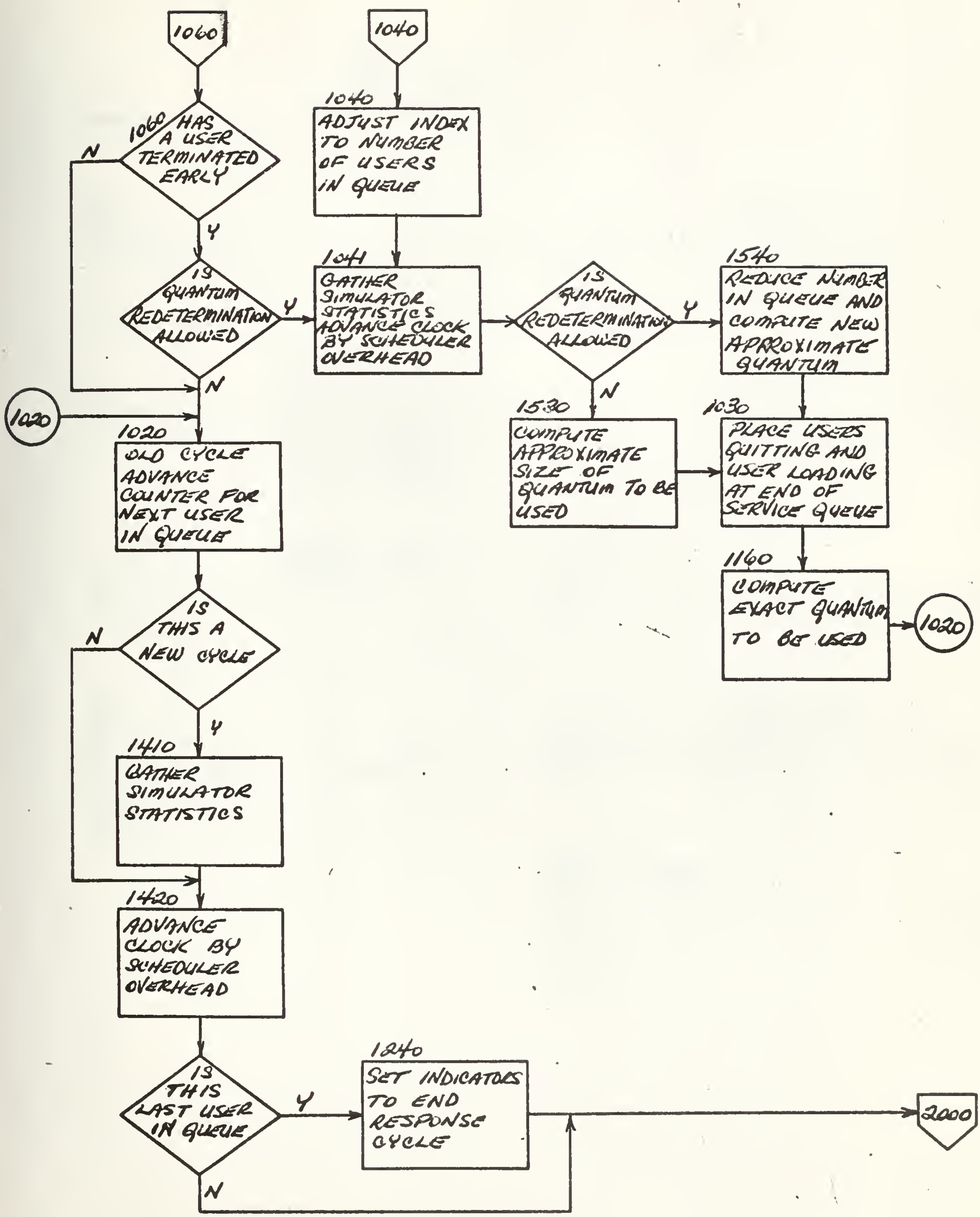
APPENDIX II

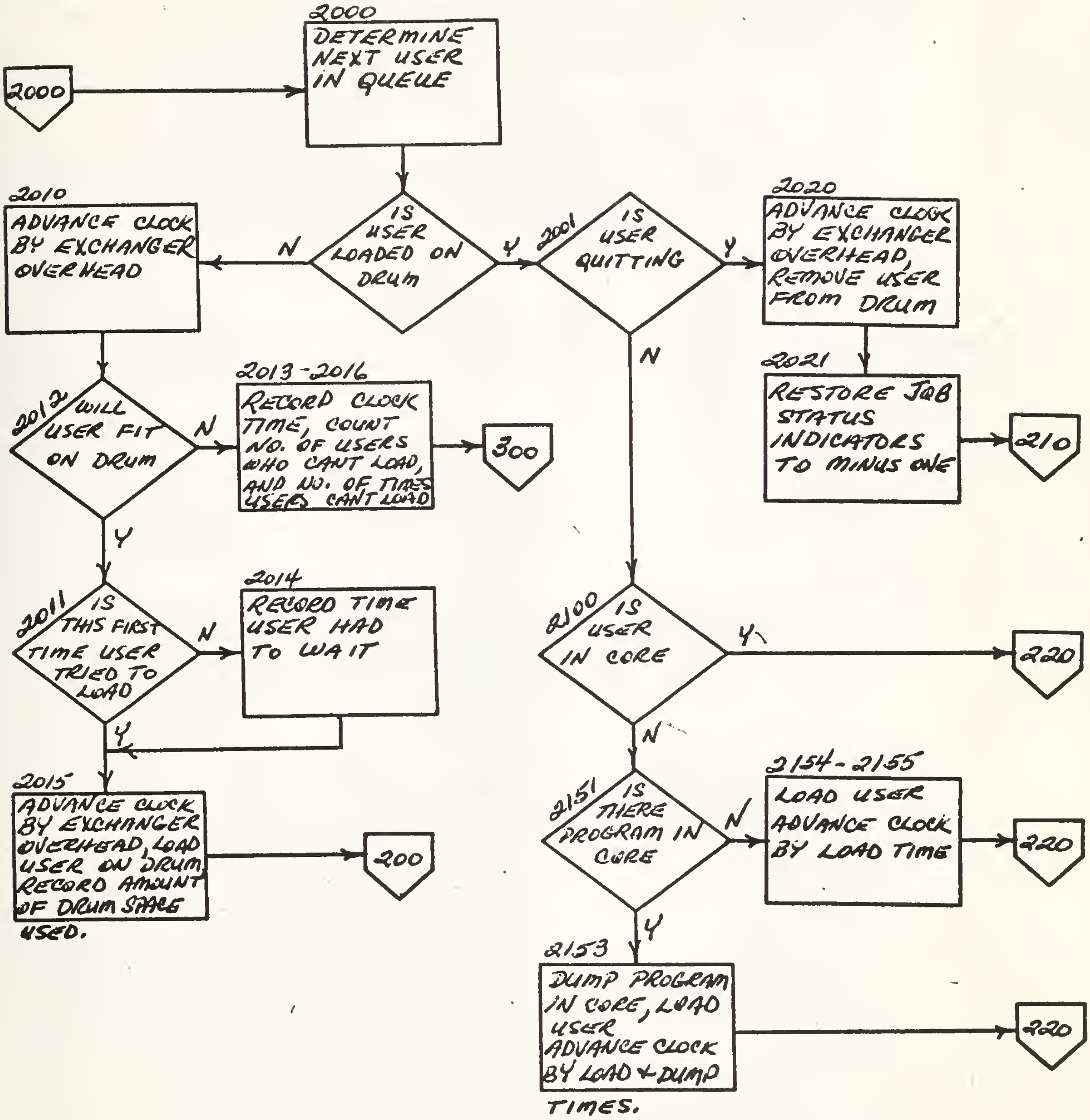
PROGRAM SIM

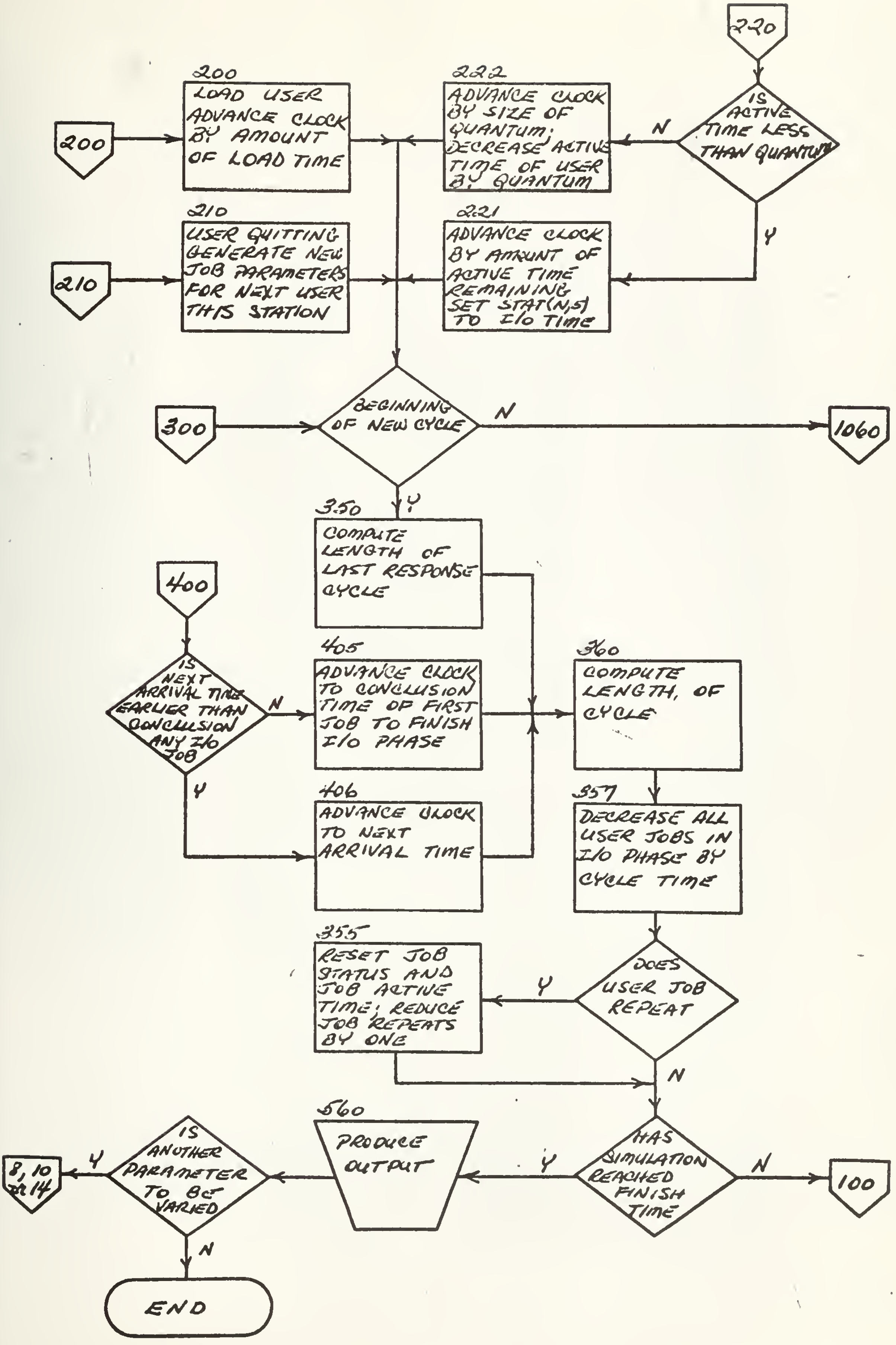


NUMBERS WITHIN THE CONNECTORS CORRESPOND TO STATEMENT NUMBERS IN PROGRAM SIM.


1050
1050
ADVANCE CLOCK BY TIME TO SCAN FOR USERS REQUESTING SERVICE
IS USER REQUESTING SERVICE
Y
N
1070
IS USER LOADED ON DRUM?
N
Y
1080
IS ANOTHER USER SCHEDULED TO BE LOADED
N
Y
1090
PLACE USER IN CYCLE
1010
IS USER QUITTING
Y
N
1110
PLACE USER IN QUEUE OF USERS QUITTING
JOB WILL NOT BE LOADED THIS RESPONSE CYCLE
1100
PLACE USER IN QUEUE OF USERS TO BE SERVICED
N
IS QUEUE SIZE EXCEEDED
ARE BACKGROUND USERS ALLOWED
N
Y
1440
IS A USER SCHEDULED TO LOAD
N
1400
ARE THERE ANY JOBS IN THE SERVICE QUEUE
N
1121
ARE THERE ANY JOBS QUITTING SERVICE
N
420
1442
PLACE BACKGROUND USER IN QUEUE
1040


1060
1040
1060 HAS A USER TERMINATED EARLY
N
Y
1040
ADJUST INDEX TO NUMBER OF USERS IN QUEUE
IS QUANTUM REDETERMINATION ALLOWED
Y
N
1041
GATHER SIMULATOR STATISTICS ADVANCE CLOCK BY SCHEDULER OVERHEAD
IS QUANTUM REDETERMINATION ALLOWED
Y
N
1540
REDUCE NUMBER IN QUEUE AND COMPUTE NEW APPROXIMATE QUANTUM
1530
COMPUTE APPROXIMATE SIZE OF QUANTUM TO BE USED
1030
PLACE USERS QUITTING AND USER LOADING AT END OF SERVICE QUEUE
1160
COMPUTE EXACT QUANTUM TO BE USED
1020
1020
1020
OLD CYCLE ADVANCE COUNTER FOR NEXT USER IN QUEUE
IS THIS A NEW CYCLE
N
Y
1410
GATHER SIMULATOR STATISTICS
1420
ADVANCE CLOCK BY SCHEDULER OVERHEAD
IS THIS LAST USER IN QUEUE
Y
N
1240
SET INDICATORS TO END RESPONSE CYCLE
2000


2000
2000
DETERMINE NEXT USER IN QUEUE
2010
ADVANCE CLOCK BY EXCHANGER OVERHEAD
N
IS USER LOADED ON DRUM
Y
2001
IS USER QUITTING
Y
2020
ADVANCE CLOCK BY EXCHANGER OVERHEAD, REMOVE USER FROM DRUM
N
2021
RESTORE JOB STATUS INDICATORS TO MINUS ONE
210
2012
WILL USER FIT ON DRUM
N
2013-2016
RECORD CLOCK TIME, COUNT NO. OF USERS WHO CANT LOAD, AND NO. OF TIMES USERS CANT LOAD
300
Y
2011
IS THIS FIRST TIME USER TRIED TO LOAD
N
2014
RECORD TIME USER HAD TO WAIT
Y
2015
ADVANCE CLOCK BY EXCHANGER OVERHEAD, LOAD USER ON DRUM RECORD AMOUNT OF DRUM SPACE USED.
200
2100
IS USER IN CORE
Y
220
N
2151
IS THERE PROGRAM IN CORE
N
2154-2155
LOAD USER ADVANCE CLOCK BY LOAD TIME
220
Y
2153
DUMP PROGRAM IN CORE, LOAD USER ADVANCE CLOCK BY LOAD + DUMP TIMES.
220


220
200
LOAD USER ADVANCE CLOCK BY AMOUNT OF LOAD TIME
222
ADVANCE CLOCK BY SIZE OF QUANTUM; DECREASE ACTIVE TIME OF USER BY QUANTUM
N
IS ACTIVE TIME LESS THAN QUANTUM
200
210
USER QUITTING GENERATE NEW JOB PARAMETERS FOR NEXT USER THIS STATION
221
ADVANCE CLOCK BY AMOUNT OF ACTIVE TIME REMAINING SET STAT(N,5) TO I/O TIME
Y
210
300
BEGINNING OF NEW CYCLE
N
1060
Y
350
COMPUTE LENGTH OF LAST RESPONSE CYCLE
400
IS NEXT ARRIVAL TIME EARLIER THAN CONCLUSION ANY I/O JOB
N
405
ADVANCE CLOCK TO CONCLUSION TIME OF FIRST JOB TO FINISH I/O PHASE
360
COMPUTE LENGTH, OF CYCLE
Y
406
ADVANCE CLOCK TO NEXT ARRIVAL TIME
357
DECREASE ALL USER JOBS IN I/O PHASE BY CYCLE TIME
355
RESET JOB STATUS AND JOB ACTIVE TIME; REDUCE JOB REPEATS BY ONE
Y
DOES USER JOB REPEAT
N
560
PRODUCE OUTPUT
8, 10 or 14
Y
IS ANOTHER PARAMETER TO BE VARIED
Y
HAS SIMULATION REACHED FINISH TIME
N
100
N
END

```
      PROGRAM SIM                FTN63

      DIMENSION MCORE(50,4),MDRUM(50,4)
      DIMENSION IQUIT(50),IQUE(50),STAT(50,10),OUTABL(15)
      DIMENSION TOTOUT(50,40),SAVE(50,3),DATA(10),YFREQ(20),QA(6)
      DIMENSION Y(900),Z(900), X(900),XD(900),ITITLE(12),ITITLE1(12)
      COMMON/A/ GENR(50,10) , AVERAGE(10,8) , DIST( 10)
      COMMON/B/MAXCOR,CLOCK,DST,IOUTCON,IR,JN,JC,JP,JD
    1 FORMAT(7F10.3)
    7 FORMAT(6I2,I3,I2)
   51 FORMAT (1H1,//////////)
   80 FORMAT(//2X,8HVARIABLE,6X,15HCYCLE   AVERAGE,3X,7HAVERAGE,3X,
     17HAVERAGE,24H  AVERAGE  COMPUTATIONAL,8H AVERAGE,2X,7HAVERAGE)
   81 FORMAT(2X,9HPARAMETER,5X,14HCOUNT    CYCLE,4X,17HNUMBER    QUANTUM
     1,2X,19HOVERHEAD EFFICIENCY,4X,8HEXCHANGE,1X,8HEXCHANGE)
   82 FORMAT(25X,4HTIME,5X,8HIN QUEUE,34X,8HOVERHEAD,1X,4HTIME//)
   83 FORMAT(//2X,7HAVERAGE,4X,7HMAXIMUM,3X,7HMAXIMUM,2X,7HMAXIMUM,3X,
     17HMAXIMUM,3X,7HMAXIMUM,3X,7HMAXIMUM,3X,8HSTATIONS,4X,7HAVERAGE)
   84 FORMAT(2X,8HOVERLOAD,3X,7HQUANTUM,3X,5HCYCLE,4X,6HNUMBER,4X,
     18HOVERHEAD,2X,8HOVERLOAD,2X,6HNUMBER,4X,8HSERVICED,4X,7HSERVICE)
   85 FORMAT(23X,4HTIME,5X,8HIN QUEUE,22X,8HSTATIONS,14X,4HTIME//)
   86 FORMAT(//3X,3HJOB,4X,20HCLOCK   STATION  JOB,4X,7HARRIVAL,8X,
     14HLOAD,6X,6HACTIVE,9X,3HI/O,5X,7HREPEATS,5X,4HSIZE)
   87 FORMAT(7H NUMBER,3X,4HTIME,4X,12HNUMBER  TYPE,4X,4HTIME,11X,4HTIME
     16X,4HTIME,10X,4HTIME//)
   89 FORMAT(35X,21HSIMULATOR OUTPUT DATA)
   90 FORMAT(    2F15.3,F10.6,3F15.1)
   91 FORMAT(F15.1,2F8.1,F12.6,2F9.1)
   92 FORMAT(10F9.6)
   94 FORMAT(39X,18HGENERATED JOB DATA////)
  515 FORMAT  (5X,F2.0,7X,F6.0,1X,2F10.4,F8.3,F10.4,F11.2,2X,2F10.4)
  516 FORMAT(F9.4,1X,2F10.4,3X,F4.0,4X,F8.4,3X,F5.0,7X,F3.0,5X,F6.0,5X,
     1F8.2)

C     INITIALIZE

      CLOKMAX=28800.0
      EXOVF=0.000100
      EXOVO = 0.000030
```

```
      MAXCOR=32000
      MAXDRM=400000
      MAXDSC=32000
      QA(1)=.200
      QA(2)=.150
      QA(3)=.100
      QA(4)=.075
      QA(5)=.050
      QA(6)=.025
      SOVRHD1=0.050
      SOVRHD2=0.010
      SOVRHD3=0.002
      XINCREM=250.0
      XINITIAL=250.0
      XINT=10.0
      XINC=10.0
      INTERVL=14
      X(1)=XINITIAL
      XD(1)=XINT
      INT=INTERVL-1
      DO 6001 I=1,INT
      XD(I+1)=XD(I)+XINC
 6001 X(I+1)=X(I)+XINCREM
    8 READ 1,((AVERAGE(J,I),I=1,7 ),J=1,10)
   10 IR=1
      PRINT 51
      PRINT 94
      PRINT 86
      PRINT 87
   14 READ 7,NCYCTM,ISTOPF,ISKEDTP,NU,IVARY,IFINIS,IQMAX,IOUTCON
      DO 6003 I=1,900
      Y(I)=0.0
 6003 Z(I)=0.0
      DST = 0.0
      JC=0
      JD=0
      JN = 0
      JP=0
      DO 3 J=1,10
```

```
    3 DIST(J)=0.0
      DO 4 J=1,NU
      DO 4 I=1,10
    4 STAT(J,I)=-1.0
      CLOCK=0.0
      ZERO=UNIFORM(0)

C     GENERATE FIRST NU JOBS

      DO 5 J=1,NU
      MDRUM(J,3)=0
      MCORE(J,4)=0
      IQUE(J)=0
      CALL SET(J)
    5 CONTINUE
      CORET=0.0
      COREUT=0.0
      CSOVRHD=0.0
      CYCNT=0.0
      CYTIME=0.0
      CYTIMMX=0.0
      ENDCYCL=1.0
      FINISH=0.0
      FRX=0.0
      FTCYCNT=0.0
      INU=1
      ITOVLD=0
      LASTI=0
      LDCHK=0
      MNOVLD=0
      MNQUE=0
      NDRUM=0
      NOLOAD=0
      NWCYCL=1
      QMAX=0.0
      QTOT=0.0
      REQUEST=0.0
      SOVD=0.0
      SOVRHD=0.0
```

```
      SOVRHDM=0.0
      SUMRX=0.0
      TCYTM=0.0
      TEDUMP=0.0
      TEFEXCH=0.0
      TELOAD=0.0
      TERM=0.0
      TEXCH = 0.0
      TNQUE=0.0
      TSUM=0.0
      TSUM1=0.0
      TSUM2=0.0
      TSUM3=0.0
      TSUM4=0.0
      TSUM5=0.0
      TSUM6=0.0
      WAITC=0.0
      WAITT=0.0

C     SCHEDULER SET-UP

   12 GO TO (20,21,22,23),IVARY
   20 VARPARA=IQMAX
      GO TO 25
   21 VARPARA=ISKEDTP
      GO TO 25
   22 VARPARA=NCYCTM
      GO TO 25
   23 VARPARA=NU
   25 GO TO (30,31,32),ISKEDTP

C     REGULAR ALGORITHM

   30 INET=2
      GO TO 100

C     REDETERMINATION OF QUANTUM AFTER EARLY TERMINATION NOT ALLOWED

   31 INET=1
```

```
      GO TO 100
C     BACKGROUND USERS ALLOWED
   32 GENR(1,1)=CLOCK
      GENR(1,3)=CLOKMAX
      INU=2
      INET=2
C     ARRIVAL DETERMINATION
  100 TIMERUN=0.0
      DO 104 I=1,NU
  101 IF(GENR(I,1)-CLOCK) 102,102,104
  102 STAT(I,2)=1.0
      STAT(I,9)=1.0
      GENR(I,1)=GENR(I,1)+1000000.0
  104 CONTINUE
  105 SCYCLE=CLOCK
      GO TO 1000
C     NO ACTIVE USERS
  400 ENDCYCL=1.0
      CYTIME=CLOCK-SCYCLE
      SMALLB = GENR(1,1) - CLOCK
      DO 401 I=2,NU
      IF(GENR(I,1) - CLOCK) 401,410,410
  410 IF(GENR(I,1) - CLOCK  - SMALLB) 402,401,401
  402 SMALLB = GENR(I,1) - CLOCK
  401 CONTINUE
      SMALLA = STAT(1,5)
      IF(SMALLA) 403,404,404
  403 SMALLA = 1000000.0
  404 DO 407 I=2,NU
      IF(STAT(I,5)) 407,408,408
  408 IF(SMALLA - STAT(I,5)) 407,407,409
  409 SMALLA = STAT(I,5)
```

```
  407 CONTINUE
      IF(SMALLA-SMALLB)405,406,406
  405 CLOCK=CLOCK+SMALLA
      TSUM1=TSUM1 + SMALLA
      GO TO 360
  406 CLOCK=CLOCK+SMALLB
      TSUM1=TSUM1 + SMALLB
      GO TO 360

C     USER IS LOADING

  200 N=IQUE(NEXT)
      REQUEST=REQUEST+1.0
      LDCHK=0
      SAVE(N,3)=CLOCK
      STAT(N,4)=GENR(N,3)
      CLOCK=CLOCK+GENR(N,2)
      TSUM3= TSUM3 + GENR(N,2)
      GO TO 300

C     USER IS QUITTING

  210 J=IQUE(NEXT)
      CALL SET(J)
      GO TO 300

C     USER IS ACTIVE

  220 N=IQUE(NEXT)
      IF(STAT(N,4)-Q)221,221,222
  222 STAT(N,4)=STAT(N,4)-Q
      TSUM4= TSUM4 + Q
      TIMERUN=Q
      CLOCK=CLOCK+Q
      QTOT=QTOT+Q
      FTCYCNT=FTCYCNT + 1.0
      GO TO 300

C     TERMINATION OF ACTIVE JOB WITH LESS THAN QUANTUM OF TIME
```

```
  221 TIMERUN=STAT(N,4)
      TERM=1.0
      STAT(N,9)=-1.0
      STAT(N,5)=GENR(N,4)
      CLOCK=CLOCK+TIMERUN
      TSUM4= TSUM4 + TIMERUN
      INQUE=INQUE-1
      FTCYCNT=FTCYCNT + 1.0
      QTOT=QTOT+TIMERUN
      GO TO 300

C     END OF CYCLE CHECK AND HOUSE-KEEPING
C     ENDCYCL=0.0 OR LESS IF OLD CYCLE, ENDCYCL=1.0 IF NEW CYCLE

  300 IF(ENDCYCL)1060,1060,350
  350 CYTIME=CLOCK-SCYCLE
      TCYTM=TCYTM+CYTIME
  360 ACYTIME=CLOCK-SCYCLE
      DO 351 I=1,NU
      IF(STAT(I,5))351,351,352
  352 IF(STAT(I,10))358,358,357
  358 STAT(I,10)=1.0
      GO TO 351
  357 IF(STAT(I,5)-ACYTIME)353,353,354

C     DECREASES I/O TIME FOR EACH JOB IN I/O CYCLE

  354 STAT(I,5)=STAT(I,5)-ACYTIME
      GO TO 351
  353 CONTINUE
      STAT(I,5)=-1.0
      STAT(I,10)=-1.0
      IF(GENR(I,5))356,356,355
  355 STAT(I,4)=GENR(I,3)
      STAT(I,9)=1.0
      GENR(I,5)=GENR(I,5)-1.0
      GO TO 351
  356 CONTINUE
      RX=CLOCK-SAVE(I,3)
```

```
      SUMRX=SUMRX + RX
      FINISH=FINISH+1.0
      IF(FRX-RX)361,362,362
  361 FRX=RX
  362 CONTINUE
      CALL ACCUM (INTERVL,RX,Y,X)
      STAT(I,9)=1.0
      STAT(I,6)= 1.0
  351 CONTINUE
      IF(CLOCK-CLOKMAX) 100,100,500

C     BASIC SCHEDULER
C     TERM=1.0 IF JOB HAS BEEN TERMINATED EARLY

 1000 IF(ENDCYCL)1060,1060,1050
 1060 IF(TERM)1020,1020,1430
 1430 GO TO (1020,1041),INET

C     NEW CYCLE

 1050 IT = 1
      ENDCYCL=0.0
      IQ=1
      NEXT=0
      CLOCK=CLOCK+SOVRHD3
      TSUM2 = TSUM2 + SOVRHD3
      SOVRHD=SOVRHD+SOVRHD3

C     QUEUE FORMATION FIRST PHASE

      DO 1120 IS=INU,NU
      IF(STAT(IS,9))1120,1120,1070
 1070 IF(STAT(IS,2))1010,1010,1080

C     LDCHK=0.0 OR LESS IF JOB IS TO BE LOADED, LDCHK=1.0 IF JOB IS LOADED

 1080 IF(LDCHK)1090,1090,1120
 1090 ILOD=IS
      LDCHK=1
```

```
      GO TO 1120
 1010 IF(STAT(IS,6))1100,1100,1110
 1110 IQUIT(IT)=IS
      IT=IT+1
      GO TO 1120
 1100 IQUE(IQ)=IS
      IQ=IQ+1

C     BACKGROUND USER NOT SERVICED IF IQ=IQMAX

      IF(IQ-IQMAX)1120,1120,1440
 1120 CONTINUE
      IF(ISKEDTP-3)1440,1442,1440
 1440 IF(LDCHK)1400,1400,1040

C     IF IQ=1 NO JOBS ARE LOADED

 1400 IF(IQ-1)1040,1121,1040

C     1040 FOR ANY REQUESTS 400 FOR NO REQUESTS
C     IF IT=1 NO JOBS ARE QUITTING

 1121 IF(IT-1)1040,400,1040

C     BACKGROUND USERS ALLOWED

 1442 IQUE(IQ)=1
      IQ=IQ+1

C     STATISTICS GATHERING

 1040 IQ=IQ-1
      INQUE=IQ
 1041 CLOCK=CLOCK+SOVRHD2
      TSUM2 = TSUM2 + SOVRHD2
      SOVRHD=SOVRHD+SOVRHD2
      NQUE=IQ
      NOQ=0
      DO 1570 I=1,NU
```

```
      IF(STAT(I,9))1570,1570,1250
 1250 NOQ=NOQ+1
 1570 CONTINUE
      NOVLD=NOQ-INQUE-IT+1-LDCHK
      ITOVLD=ITOVLD+NOVLD
      TOVLD=ITOVLD
      IF(CYCNT)3000,3000,3001
 3000 AOVLD = 0.0
      GO TO 2999
 3001 AOVLD=TOVLD/CYCNT
 2999 IF(MNOVLD-NOVLD)1381,1381,1390
 1381 MNOVLD=NOVLD
 1390 FMNOVLD=MNOVLD

C     EARLY TERMINATION
C     IF TERM=0.0 OR LESS DETERMINE QUANTUM FOR CYCLE, TERM=1.0 REDETERMINE
C     QUANTUM FOR REMAINDER OF CYCLE

      IF(TERM)1530,1530,1540
 1540 FNQUE=NQUE-NEXT
      FNEXT=NEXT
      FNCYCTM=NCYCTM
      FNCYCTM=FNCYCTM-(SOVRHD+(FNEXT-1.0)*Q+TIMERUN+((TEDUMP+TELOAD)*
     1FNEXT)/FTCYCNT)
      Q=FNCYCTM/FNQUE
      GO TO 1550

C     QUANTUM DETERMINATION FIRST PHASE

 1530 FNCYCTM=NCYCTM
      FNQUE=NQUE
      Q=FNCYCTM/FNQUE
 1220 NWCYCL=1
 1550 IF(IT-1)1500,1500,1030

C     QUEUE FORMATION SECOND PHASE

 1030 DO 1130 I=2,IT
 1130 IQUE(NQUE+IT-1)=IQUIT(IT-1)
```

```
 1500 IF(LDCHK)1520,1520,1510
 1510 IQUE(NQUE+IT)=ILOD
 1520 CONTINUE

C     QUANTUM DETERMINATION SECOND PHASE

      DO 1150 KJ=1,5
      IF(Q-QA(KJ))1150,1160,1160
 1160 Q=QA(KJ)
      GO TO 1020
 1150 CONTINUE
      Q=QA(6)

C     OLD CYCLE

 1020 NEXT=NEXT+1
      TERM=-1.0

C     NWCYCL=1 GATHER STATISTICS, IF NWCYCL DOESNT EQUAL 1 BYPASS STATS GATHER

      IF(NWCYCL)1420,1420,1410

C     STATISTICS GATHERING

 1410 CYCNT=CYCNT+1.0
      TNQUE=TNQUE+FNQUE
      IF(QMAX-Q)1300,1300,1310
 1300 QMAX=Q
 1310 IF(CYTIMMX-CYTIME)1320,1320,1330
 1320 CYTIMMX=CYTIME
 1330 IF(MNQUE-NQUE)1340,1340,1370
 1340 MNQUE=NQUE
 1370 FMNQUE=MNQUE
 1420 CLOCK=CLOCK+SOVRHD1
      TSUM2 = TSUM2 + SOVRHD1
      NWCYCL=0
      SOVRHD=SOVRHD+SOVRHD1

C      IF(NEXT-NQUE-IT+1-LDCHK) EQUALS 0 LAST USER IN CYCLE IS TO BE SERVICED
```

```
      IF(NEXT-NQUE-IT+1-LDCHK)2000,1240,1240
 1240 ENDCYCL=1.0
      LDCHK=0
 1350 IF(SOVRHDM-SOVRHD)1360,1360,1380
 1360 SOVRHDM=SOVRHD
 1380 CSOVRHD=CSOVRHD+SOVRHD
      SOVRHD=0.0

C     BASIC EXCHANGE PACKAGE

 2000 I = IQUE(NEXT)
      SWPOVD = 0.000300

C     IF GREATER THAN 0 CHANNEL I NEEDS LOADING
C     STAT(I,2)=1.0 IF USER IS TO BE LOADED ON DRUM, STAT(I,2)=-1.0 IF USER IS
C     LOADED ON DRUM

      IF(STAT(I,2))2001,2001,2010
 2010 CLOCK = CLOCK + EXOVO
      TSUM6= TSUM6 + EXOVO
      SOVD = SOVD + EXOVO
      MSIZE=GENR(I,6)

C     TEST TO DETERMINE IF DRUM CAPACITY IS EXCEEDED

      IF(MSIZE + NDRUM - MAXDRM) 2011,2011,2012

C     NO LOAD
C     MDRUM(I,3)=-1.0 IF PROGRAM COULD NOT BE LOADED PREVIOUSLY
C     MDRUM(I,3)=0.0 OR GREATER IF FIRST TIME PROGRAM COULD NOT BE LOADED

 2012 IF(MDRUM(I,3))2016,2013,2013
 2013 CONTINUE
      MDRUM(I,3) = -1
      SAVE(I,1) = CLOCK

C     WAITC COUNTS NUMBER OF USERS WHO MUST WAIT BEFORE BEING LOADED

      WAITC = WAITC + 1.0
```

```
C     NOLOAD COUNTS NUMBER OF TIMES USER MUST WAIT BEFORE BEING LOADED

 2016 NOLOAD = NOLOAD + 1
      GO TO 300

C     LOAD OK
C     MDRUM(I,3)=-1.0 IF PROGRAM COULD NOT BE LOADED PREVIOUSLY
C     MDRUM(I,3)=0.0 IF PROGRAM CAN BE LOADED IMMEDIATELY

 2011 IF(MDRUM(I,3)) 2014,2015,2015

C      WAITT ACCUMULATES TIME USERS MUST WAIT BEFORE BEING LOADED

 2014 WAITT = WAITT + ( CLOCK - SAVE(I,1))

C     LOAD JOB ON DRUM

 2015 MDRUM(I,3) = MSIZE
      NDRUM = NDRUM + MSIZE
      STAT(I,2)=-1.0
      STAT(I,8)=GENR(I,6)
      CLOCK = CLOCK + EXOVF
      TSUM6 = TSUM6 + EXOVF
      GO TO 200
 2001 IF(STAT(I,6)) 2002,2002,2020

C     IF GREATER THAN 0 CHANNEL I IS QUITTING
C     REMOVE JOB FROM DRUM

 2020 MSIZE= STAT(I,8)
      NDRUM=NDRUM - MSIZE
      CLOCK=CLOCK+EXOVO
      TSUM6= TSUM6 + EXOVO
      SOVD=SOVD + EXOVO
      DO 2021 IA =1,9
 2021 STAT(I,IA) = -1.0
      GO TO 210
```

```
C     NORMAL EXCHANGE
C     EXCHANGE METHOD 1 NO CON

 2002 MSIZE=MDRUM(I,3)
      CORET = CORET + 1.0
      COREUT = COREUT + STAT(I,8)
 2100 IF(MCORE(I,4)) 2151,2151,2150

C     PROGRAM IS IN CORE

 2150 TLOAD = 0.0
      TDUMP=0.0
      GO TO 2152
 2151 IF(LASTI) 2154,2154,2153
 2154 TDUMP=0.0
      GO TO 2155

C     EXCHANGE LAST USER

 2153 TDUMP = STAT(LASTI,8) * 0.000003
      TEDUMP = TDUMP + TEDUMP

C     MCORE(LASTI,4)=-1 IF PROGRAM IS ON DRUM BUT NOT IN CORE

      MCORE(LASTI,4) =-1
 2155 TLOAD = STAT(I,8) * 0.000003
      TELOAD = TLOAD + TELOAD

C     MCORE(I,4)=1 IF PROGRAM IS ON DRUM AND IS LOADED IN CORE

      MCORE(I,4) = 1

C     ADVANCE CLOCK CORRESPONDING TO EXCHANGE TIMES

 2152 TEXCH = TEXCH + TDUMP + TLOAD + SWPOVD
      TEFEXCH=TELOAD+TEDUMP+SWPOVD
      SOVD = SOVD + SWPOVD
```

```
      SWPOVD = 0.0
      CLOCK = CLOCK + TLOAD + TDUMP
      TSUM5=TSUM5 + TLOAD + TDUMP

C     LASTI INDICATES LAST USER DURING RESPONSE CYCLE

      LASTI = I
      GO TO 220
  500 CONTINUE

C     OUTPUT

      IF(WAITC) 559,559,560
  559 WAITC = 1.0
  560 CONTINUE
      AWAIT = WAITT/ WAITC
      COREF = COREUT/ CORET
      EXEFF = (TEXCH - TEFEXCH)/TEXCH
      ASWOVD=SOVD/CYCNT
      ASWAP=TEXCH/CYCNT
      AOVRHD=CSOVRHD/CYCNT
      CYTMAVE=TCYTM/CYCNT
      AVNQUE=TNQUE/CYCNT
      QAVE=QTOT/FTCYCNT
      COMPEFF=(QTOT/TCYTM) * 100.0
      DO 551 I=1,10
  551 DATA(I) = DIST(I)/DST
   11 TOTOUT(IR,1)=VARPARA
      TOTOUT(IR,2)=CYCNT
      TOTOUT(IR,3)=CYTMAVE
      TOTOUT(IR,4)=AVNQUE
      TOTOUT(IR,5)=QAVE
      TOTOUT(IR,6)=AOVRHD
      TOTOUT(IR,7)=COMPEFF
      TOTOUT(IR,8)=AOVLD
      TOTOUT(IR,9)=QMAX
      TOTOUT(IR,10)=CYTIMMX
      TOTOUT(IR,11)=FMNQUE
      TOTOUT(IR,12)=SOVRHDM
```

```
      TOTOUT(IR,13)=FMNOVLD
      TOTOUT(IR,14)=NU
      TOTOUT(IR,15)=TEXCH
      TOTOUT(IR,16)=TEFEXCH
      TOTOUT(IR,17)=EXEFF
      TOTOUT(IR,18)=COREF
      TOTOUT(IR,19)=CORET
      TOTOUT(IR,20)=FRX
      TOTOUT(IR,21)=AWAIT
      TOTOUT(IR,22)=NOLOAD
      TOTOUT(IR,23)=DST
      TOTOUT(IR,24)=SOVD
      TOTOUT(IR,25)=REQUEST
      TOTOUT(IR,26)=TCYTM
      TOTOUT(IR,27)=ASWOVD
      TOTOUT(IR,28)=ASWAP
      TOTOUT(IR,29)=FINISH
      TOTOUT(IR,30)=SUMRX/FINISH
      DO 571 IM=1,10
      IN = IM + 30
      TOTOUT(IR,IN) = DATA(IM)
  571 CONTINUE
      TSUM=TSUM1+TSUM2+TSUM3+TSUM4+TSUM5+TSUM6
      PRINT 4000,TSUM,TSUM1,TSUM2,TSUM3,TSUM4,TSUM5,TSUM6
 4000 FORMAT (9X,7F12.5)
      CALL HIST (Y,X,INTERVL,ITITLE)
      IF(ISTOPF)579,579,574
  574 PRINT 51
      PRINT 89
      PRINT 80
      PRINT 81
      PRINT 82
      PRINT 515,(((TOTOUT(I,J),J=1,7),(TOTOUT(I,J),J=27,28)),I=1,IR)
      PRINT 83
      PRINT 84
      PRINT 85
      PRINT 516,(((TOTOUT(I,J),J=8,14),(TOTOUT(I,J),J=29,30)),I=1,IR)
      PRINT 51
      PRINT 90, ((TOTOUT(IT,IS),IS=15,20),IT=1,IR)
```

```
      PRINT 91, ((TOTOUT(IT,IS),IS=21,26),IT=1,IR)
      PRINT 92,((TOTOUT(IT,IS),IS=31,40),IT=1,IR)
      GO TO 572
  579 IR=IR+1
      GO TO 14
  572 IF(IFINIS) 10,8,558
  558 CONTINUE
      STOP
      END
```

```
      SUBROUTINE ACCUM (INTERVL,RX,YFREQ,X)
      DIMENSION YFREQ(900),X(900)
      IF(RX.LE.X(1))2,3
    2 YFREQ(1)=YFREQ(1)+1.0
      GO TO 10
    3 DO 5 I=2,INTERVL
      IF(RX.LE.X(I))4,5
    5 CONTINUE
      YFREQ(INTERVL)=YFREQ(INTERVL)+1.0
      GO TO 10
    4 YFREQ(I)=YFREQ(I)+1.0
   10 RETURN
      END
```

```
      SUBROUTINE HIST (YFREQ,X,INTERVL,ITITLE)
      DIMENSION YFREQ(900),ANEWY(900),AX(900),X(900),ITITLE(12)
      NEWINT=INTERVL * 2 + 1
      DO 3 I=1,900
      AX(I)=0.0
    3 ANEWY(I)=0.0
      DO 1 I=1,INTERVL
      AX(2*I)=X(I)
    1 ANEWY(2*I)  =YFREQ(I)
      NN=INTERVL*2
      DO 2 I=2,NN,2
      AX(I+1)=AX(I)
    2 ANEWY(I-1)=ANEWY(I)
      ANEWY(NEWINT)=0.0
      PRINT 5
    5 FORMAT (1H1)
      PRINT 4, (AX(I),ANEWY(I),I=1,29)
    4 FORMAT     (10X,F5.0,5X,F5.0)
      CALL DRAW(NEWINT,AX,ANEWY,0,0,LQ,ITITLE,0,0,0,0,0,0,6,8,0,LAST)
      RETURN
      END
```

```
C      JOB GENERATOR

       SUBROUTINE SET(J)
       DIMENSION UNUM(6)
       COMMON/A/ GENR(50,10) , AVERAGE(10,8) , DIST( 10)
       COMMON/B/MAXCOR,CLOCK,DST,IOUTCON,IR,JN,JC,JP,JD
    51 FORMAT (1H1,////////)
   615 FORMAT(1X,I4,F10.1,2X,2I6,5F12.4,I9)
       I=0
       JN=JN+1
       TEMP=0.0
       DST = DST + 1.0
       DO 600 IC=1,5
       GAUSRN=0.0
       DO 20 JZ=1,12
    20 GAUSRN=GAUSRN+UNIFORM(JZ)
   600 UNUM(IC)=GAUSRN/6.0

C GENERATE A NEW UNIRN FOR PROBABILITY OF JOB

       UNIRN=UNIFORM(6)
   612 I=I+1
       TEMP=TEMP+AVERAGE(I,7)
       IF(UNIRN.LT.TEMP)610,611
   611 IF(I.LT.10)612,610
   610 DIST(I)=DIST(I)+1.0

C ARRIVAL TIME

   614 UNIRN=UNIFORM(I)
       GENR(J,1)=CLOCK+LOGF(1.0/(1.0-UNIRN))*AVERAGE(I,1)

C ACTIVE TIME

       GENR(J,3)=AVERAGE(I,3)*UNUM(2)

C I/O TIME

       GENR(J,4)=AVERAGE(I,4)*UNUM(3)
```

```
C REPEATS

      GENR(J,5)=AVERAGE(I,5)*UNUM(4)

C SIZE

      GENR(J,6)=AVERAGE(I,6)*UNUM(5)

C DETERMINE IF PROGRAM SIZE EXCEEDS CORE SIZE

  620 MSIZE = GENR(J,6)
      IF(MSIZE.LT.MAXCOR)618,619
  619 GENR(J,6)=GENR(J,6)-1000.0
      GO TO 620

C LOAD TIME DETERMINATION(LOAD FROM DISC TO DRUM)

  618 ACCESS=MSIZE/8000+1
      GENR(J,2)=ACCESS*UNUM(1)*0.14+GENR(J,6)*12.0E-6
      IF(IOUTCON.EQ.4)621,616
  621 IF(IR-1)616,622,617
  622 JP=JP+1
      IF(JP.LT.31)617,623
  623 JC=JC+1
      IF(JC.LT.41)624,625
  624 IF(JC.EQ.1)627,617
  625 IF(JD.EQ.0)626,617
  626 JD=1
  627 PRINT 51
  617 PRINT 615,JN,CLOCK,J,I,(GENR(J,K),K=1,5),MSIZE
  616 RETURN
      END
```

```
         IDENT       UNIFORM                UNIFORM(0,1) RANDOM NO. GEN.     UNIF000
         ENTRY       UNIFORM                WRITTEN AS A FUNCTION SUBPROG-   UNIF001
R        OCT         5402033450727422        RAM  WITH A SINGLE DUMMY        UNIF002
UNIFORM  SLJ         **                     ARGUMENT.                        UNIF003
         SIU     1   EXIT                                                    UNIF0035
         LIU     1   *-1                                                     UNIF004
         SIU     1   *+2                                                     UNIF0045
         INI     1   +1                                                      UNIF005
         SIL     1   EXIT                                                    UNIF0055
         LDA         **                                                      UNIF006
         ALS         +24                    IN CALLING PROGRAM, SET DUMMY=0  UNIF0065
         SAU         *+1                    TO RESET GENERATOR.  SET DUMMY   UNIF007
+        LDA         **                     EQUAL TO ANYTHING BUT ZERO TO    UNIF0071
         AJP     N   GO                     CONTINUE RANDOM SEQUENCE.        UNIF0074
         LDA         =05402033450727422        MODIFIED 14JULY65  RHS        UNIF0075
         STA         R                                                       UNIF0076
GO       ENQ         0                                                      *UNIF008
         LDA         R                                                       UNIF009
         LLS         +19                                                     UNIF010
         SCL         =0400000000000000                                       UNIF011
         ADD         R                                                       UNIF012
         ADD         R                                                       UNIF013
         ADD         R                                                       UNIF014
         STA         R                                                       UNIF015
         ARS         11                                                      UNIF016
         ADD         =0200000000000000                                       UNIF017
         FAD         =0200000000000000                                       UNIF018
EXIT     ENI     1   **                                                     *UNIF019
         SLJ         **                                                      UNIF0195
         END                                                                 UNIF020
         END
         FINIS
-EXECUTE.
```

APPENDIX IV

GLOSSARY OF FORTRAN NAMES USED IN PROGRAM SIM

ACCESS - Number of physical accesses necessary to load a job from disc to drum.

ACCUM - Subroutine to accumulate frequencies for a histogram.

ACYTIME - All cycle time whether users are active or not.

AOVLD - Average NOVLD.

AOVRHD - Average scheduler overhead.

ASWAP - Average swap time.

ASWOVD - Average swap overhead.

AVERAGE - Array of job profiles.

AVNQUE - Average number in queue.

AWAIT - Average wait time before space is available to load job on drum.

CLOCK - Accumulated time within simulation.

CLOKMAX - Length of time in seconds for which simulation is to be run.

COMPEFF - Computational efficiency.

COREF - Average size of program swapped from drum to core.

CORET - Counts number of core transfers.

COREUT - Accumulates amount of core used by all programs during simulation.

CSOVRHD - Sums SOVRHD.

CYCNT - Counts number of response cycles.

CYTIME - Length of time of a specific response cycle during which users are active.

CYTIMMX - Maximum response cycle.

CYTMAVE - Average cycle time.

DIST - Array to accumulate distribution of job types.

DST - Counts number of jobs generated.

ENDCYCL - Index indicating whether response cycle is an old or new cycle.

EXEFF - Exchange efficiency.

EXOVF - Exchange overhead to load user on drum initially.

EXOVO - Exchange overhead to load user on drum, to swap between drum and core, and to remove user from drum upon completion of service.

FINISH - Counts number of users who complete service during simulation.

FMNOVLD - MNOVLD as floating-point variable.

FMNQUE - MNQUE as floating-point variable.

FNCYCTM - NCYCTM as floating-point variable.

FNQUE - NQUE as floating-point variable.

FRX - Maximum time user who has completed service was in the system.

FTCYCNT - Counts number of quanta of service given.

GENR - Array of job parameters.

HIST - Subroutine to compute points to draw a histogram.

IFINIS - Index to control reading new job profile data, reading additional input cards to vary parameters, and to end program.

ILOD - Indicates number of station that is to be loaded during response cycle.

INET - Index used to indicate whether redetermination of quantum is allowed or not allowed.

INQUE - Number in queue after a user has completed being serviced.

INTERVL - Number of intervals in histogram.

INU - Index used to indicate whether background users are allowed or not.

IOUTCON - Index to control whether jobs generated will be printed out or not.

IQ - Index indicating number of users in queue who are to receive a quantum of service.

IQMAX - Maximum size of queue.

IQUE - Array used to store station number of users in the queue.

IQUIT - Array of users quitting service.

IR - Index that counts number of different runs for a group of data. Reset to 1 when basic input data changes.

ISKEDTP - Index to indicate which schedule algorithm is to be used.

ISTOPF - Index to control printing of data and reading of additional input cards to vary parameters.

IT - Index indicating number of users that are quitting service during a response cycle time.

ITOVLD - Accumulates number of overloads (NOVLD).

IVARY - Index to indicate the parameter to be varied.

JC - Index to control lines of printing per page for jobs generated.

JD - Index to control lines of printing per page for jobs generated.

JN - Job number.

JP - Index to control printing of jobs generated.

LASTI - Indicates previous user in the response cycle.

LDCHK - Indicates whether job is to be loaded or not.

MAXCOR - Maximum size of core available for users programs.

MAXDRM - Maximum drum storage available.

MAXDSC - Maximum disc storage available.

MCORE - Indicates whether program is loaded in core or not.

MDRUM - Indicates whether this is first time or not that program could not be loaded.

MNOVLD - Maximum value of NOVLD.

MNQUE - Maximum number of users in any response cycle.

MSIZE - Size of user job.

NCYCTM - Target response cycle time.

NDRUM - Total amount of drum space utilized for storing users' programs.

NEWINT - Number of points to be plotted to get histogram.

NEXT - Index indicating current user being serviced, or to be serviced.

NOLOAD - Number of times users are not able to load because drum is full.

NOQ - Number of stations requesting service.

NOVLD - Number of users who could not load because another user was loading during that response cycle.

NQUE - Number in queue.

NU - Number of remote stations allowed.

NWCYCL - Indicates whether cycle is a new or an old cycle.

Q - Quantum of time.

QA - Array of quantum sizes.

QAVE - Average quantum.

QMAX - Maximum quantum offered to any user.

QTOT - Accumulates quantum times.

REQUEST - Counts number of users loaded into the system.

RX - Amount of time a specific user took to complete service in the system.

SAVE(I,1) - Records time the I-th user first tries to load on drum but can not because drum is full.

SAVE(N,3) - Records time the N-th user program enters system.

SCYCLE - Time of start of response cycle.

SET - Subroutine to create job parameters.

SMALLA - Smallest I/O time remaining.

SMALLB - Nearest arrival time.

SOVD - Sums EXOVO, EXOVF, and SWPOVD.

SOVRHD - Cumulative scheduler overhead, accumulates SOVRHD1, SOVRHD2, and SOVRHD3 during each response cycle.

SOVRHD1 - Scheduler overhead between jobs.

SOVRHD2 - Scheduler overhead to do system accounting and statistics gathering.

SOVRHD3 - Scheduler overhead to scan stations for user requesting service.

SOVRHDM - Maximum value of SOVRHD.

STAT - Array of indicators of status of job being serviced.

STAT(I,2) - Indicates whether the I-th user is to be loaded on drum or is already loaded on drum.

STAT(I,4) - Active time for I-th user.

STAT(I,5) - I/O time for the I-th user.

STAT(I,6) - Indicates whether the I-th user is quitting service.

STAT(I,8) - Size of the I-th user's job.

STAT(I,9) - Indicates the number of the I-th station requesting service.

STAT(I,10)- Indicates whether the I-th user's job is in I/O phase.

SUMRX - Accumulates total time users are in the system.

SWPOVD - Swap overhead.

TCYTM - Total cycle time, accumulates CYTIME.

TDUMP - Time to dump program from core to drum.

TEDUMP - Accumulates total time to dump program from core to drum.

TEFEXCH - Total effective time to exchange programs.

TELOAD - Accumulates total time to load program from drum to core.

TEMP - Determines which job type is to be generated.

TEXCH - Exchange time.

TERM - Indicates whether or not a job has terminated early.

TIMERUN - Quantum of time job has run.

TLOAD - Time to load program from drum to core.

TNQUE - Accumulates total number of users in all queues.

TOTOUT - Array to accumulate output statistics for simulator.

TOVLD - ITOVLD as floating point variable.

TSUM - Sums TSUM1 to TSUM6; equals total time simulation has run.

TSUM1 - Sums SMALLA and SMALLB.

TSUM2 - Sums SOVRHD1, SOVRHD2, and SOVRHD3.

TSUM3 - Sums GENR(N,2)-user load time.

TSUM4 - Sums Q and TIMERUN for each job offered a quantum of service.

TSUM5 - Sums TLOAD and TDUMP.

TSUM6 - Sums EXOVO and EXOVF.

UNIFORM - Subroutine to generate a uniform random number.

UNIRN - Uniform random number.

VARPARA - Variable parameter.

WAITC - Wait count, accumulates number of users who must wait because drum capacity is full.

WAITT - Wait time, accumulates amount of time users, who can not get loaded because drum capacity is full, must wait before being loaded.

X(I) - Array of intervals in histogram.

XD(I) - Array of intervals in histogram.

XINCREM - Size of interval in histogram.

XINC - Size of interval in histogram.

XINITIAL - Distance of first interval from origin.

XINT - Distance of first interval from origin.

Y(I) - Array to accumulate frequencies in histogram.

Z(I) - Array to accumulate frequencies in histogram.

ZERO - Method to initialize uniform random number generator.

APPENDIX V

RESULTS OF SIMULATIONS

The data in this appendix are the results of the simulation runs discussed in Section 3. The results are numbered so that they may be easily correlated to the sections to which they apply, i.e., Figure 3.3.2 is the second figure referenced in Section 3.3. Each computer print out and histogram are also annotated with the pertinent data which was varied for that print out and histogram. The abscissa of the histograms is the total time a user spends in the system before his program is completed. The ordinate of the histogram is the number of users who have completed service. The last bar of the histogram is an accumulation of all users time intervals beyond the next to last interval. The small diagram above the main histogram is a breakdown of the percentage allocation of total time. The ordinate axis ranges from 0 - 80%. The first bar (I) represents percentage of idle time, the second bar (O) is the overhead time percentage, the third bar (A) is the actual service time percentage and the last bar (S) the swap time percentage.

FIGURE 3.2.1

SIMUALTOR OUTPUT DATA
WITH EARLY TERMINATIONS ALLOWED,
NOT ALLOWED AND BACKGROUND JOBS PERMITTED.
QUANTUM DETERMINATION METHOD Q_1, SCHOOL DATA, AND 20 STATIONS

VARIABLE PARAMETER	CYCLE COUNT	AVERAGE CYCLE TIME	AVERAGE NUMBER IN QUEUE	AVERAGE QUANTUM	AVERAGE OVERHEAD	COMPUTATIONAL EFFICIENCY	AVERAGE EXCHANGE OVERHEAD	AVERAGE EXCHANGE TIME
1.0000	6661.0000	2.1563	10.7098	.1319	.5850	65.4995	.0034	.0857
2.0000	6633.0000	2.1654	10.5162	.1369	.5685	66.4981	.0033	.0839
3.0000	6663.0000	2.1615	11.7214	.1250	.6081	67.7831	.0035	.0848

AVERAGE OVERLOAD	MAXIMUM QUANTUM	MAXIMUM CYCLE TIME	MAXIMUM NUMBER IN QUEUE	MAXIMUM OVERHEAD	MAXIMUM OVERLOAD	MAXIMUM NUMBER STATIONS	STATIONS SERVICED
.0017	.2000	3.1002	19.0000	.9820	1.0000	20.0000	314.0000
.0018	.2000	3.0257	19.0000	.9620	1.0000	20.0000	318.0000
.0024	.2000	2.9866	19.0000	.9920	1.0000	20.0000	277.0000

FIGURE 3.2.2

SIMUALTOR OUTPUT DATA
WITH EARLY TERMINATIONS ALLOWED,
NOT ALLOWED AND BACKGROUND JOBS PERMITTED.

QUANTUM DETERMINATION METHOD Q_2, SCHOOL DATA, AND 20 STATIONS

VARIABLE PARAMETER	CYCLE COUNT	AVERAGE CYCLE TIME	AVERAGE NUMBER IN QUEUE	AVERAGE QUANTUM	AVERAGE OVERHEAD	COMPUTATIONAL EFFICIENCY	AVERAGE EXCHANGE OVERHEAD	AVERAGE EXCHANGE TIME
1.0000	9109.0000	1.5777	13.0476	.0584	.6868	48.3051	.0040	.1096
2.0000	7768.0000	1.8493	11.7630	.0928	.6237	59.0275	.0037	.0959
3.0000	9408.0000	1.5307	13.8369	.0520	.7089	47.0152	.0042	.1028

AVERAGE OVERLOAD	MAXIMUM QUANTUM	MAXIMUM CYCLE TIME	MAXIMUM NUMBER IN QUEUE	MAXIMUM OVERHEAD	MAXIMUM OVERLOAD	MAXIMUM NUMBER STATIONS	STATIONS SERVICED
.0005	.2000	2.2694	19.0000	.9920	1.0000	20.0000	225.0000
.0039	.2000	2.5250	19.0000	1.0120	1.0000	20.0000	273.0000
.0005	.2000	2.1938	19.0000	.9820	1.0000	20.0000	199.0000

FIGURE 3.2.3

```
                         SIMUALTOR OUTPUT DATA
                    WITH EARLY TERMINATIONS ALLOWED,
                 NOT ALLOWED AND BACKGROUND JOBS PERMITTED.
           QUANTUM DETERMINATION METHOD Q3, SCHOOL DATA, AND 20 STATIONS

 VARIABLE     CYCLE    AVERAGE   AVERAGE   AVERAGE   AVERAGE COMPUTATIONAL AVERAGE   AVERAGE
PARAMETER     COUNT     CYCLE    NUMBER    QUANTUM  OVERHEAD EFFICIENCY    EXCHANGE  EXCHANGE
                        TIME    IN QUEUE                                   OVERHEAD   TIME

  1.0000 5386.0000    2.6613   11.2846     .1646     .6178   69.8053      .0036     .0916

  2.0000 5045.0000    2.8422   11.0432     .1810     .6097   70.3263      .0035     .0913

  3.0000 5749.0000    2.5049   11.7471     .1535     .6117   71.9978      .0035     .0848



 AVERAGE    MAXIMUM   MAXIMUM   MAXIMUM   MAXIMUM   MAXIMUM   MAXIMUM   STATIONS
OVERLOAD    QUANTUM    CYCLE    NUMBER    OVERHEAD  OVERLOAD   NUMBER   SERVICED
                       TIME    IN QUEUE                       STATIONS

   .0039     2.0000    5.0131   19.0000    1.0220    1.0000   20.0000   336.0000

   .0010     2.0000    6.1699   19.0000     .9620    1.0000   20.0000   342.0000

   .0063     2.0000    3.2703   19.0000    1.0120    1.0000   20.0000   301.0000
```

FIGURE 3.2.4

SIMUALTOR OUTPUT DATA
WITH EARLY TERMINATIONS ALLOWED,
NOT ALLOWED AND BACKGROUND JOBS PERMITTED.
QUANTUM DETERMINATION METHOD Q_4, SCHOOL DATA, AND 20 STATIONS

```
 VARIABLE      CYCLE   AVERAGE   AVERAGE    AVERAGE   AVERAGE COMPUTATIONAL AVERAGE  AVERAGE
PARAMETER      COUNT    CYCLE    NUMBER     QUANTUM   OVERHEAD EFFICIENCY   EXCHANGE EXCHANGE
                        TIME    IN QUEUE                                    OVERHEAD  TIME

   1.0000 5695.0000    2.5190   10.4976      .1705     .5703   71.0529       .0033    .0869

   2.0000 5335.0000    2.6913    9.8122      .1968     .5432   71.7489       .0032    .0806

   3.0000 5881.0000    2.4488   10.9100      .1646     .5700   73.3311       .0033    .0779



 AVERAGE     MAXIMUM   MAXIMUM   MAXIMUM    MAXIMUM   MAXIMUM   MAXIMUM     STATIONS
 OVERLOAD    QUANTUM    CYCLE    NUMBER     OVERHEAD  OVERLOAD  NUMBER      SERVICED
                        TIME    IN QUEUE                        STATIONS

    .0026      .2000   4.7535   19.0000      .9920    1.0000   20.0000    338.0000

    .0011      .2000   4.9812   19.0000      .9620    1.0000   20.0000    349.0000

    .0036      .2000   4.6588   18.0000     1.0020    1.0000   20.0000    309.0000
```

FIGURE 3.2.5

SIMUALTOR OUTPUT DATA
WITH EARLY TERMINATIONS ALLOWED,
NOT ALLOWED AND BACKGROUND JOBS PERMITTED.
QUANTUM DETERMINATION METHOD Q_1, LARGE JOB DATA, AND 20 STATIONS

VARIABLE PARAMETER	CYCLE COUNT	AVERAGE CYCLE TIME	AVERAGE NUMBER IN QUEUE	AVERAGE QUANTUM	AVERAGE OVERHEAD	COMPUTATIONAL EFFICIENCY	AVERAGE EXCHANGE OVERHEAD	AVERAGE EXCHANGE TIME
1.0000	4833.0000	2.9753	13.1455	.1059	.7100	46.7714	.0041	.8009
2.0000	4620.0000	3.1121	12.7277	.1199	.6899	49.0448	.0040	.7960
3.0000	5105.0000	2.8211	13.4400	.1033	.6944	49.2030	.0040	.7291

AVERAGE OVERLOAD	MAXIMUM QUANTUM	MAXIMUM CYCLE TIME	MAXIMUM NUMBER IN QUEUE	MAXIMUM OVERHEAD	MAXIMUM OVERLOAD	MAXIMUM NUMBER STATIONS	STATIONS SERVICED
.0041	.2000	4.6412	19.0000	1.0120	1.0000	20.0000	195.0000
.0019	.2000	4.6627	19.0000	.9620	2.0000	20.0000	213.0000
.0020	.2000	4.5623	19.0000	1.0220	1.0000	20.0000	182.0000

FIGURE 3.2.6

SIMUALTOR OUTPUT DATA
WITH EARLY TERMINATIONS ALLOWED,
NOT ALLOWED AND BACKGROUND JOBS PERMITTED.
QUANTUM DETERMINATION METHOD Q_2, LARGE JOB DATA, AND 20 STATIONS

```
 VARIABLE      CYCLE    AVERAGE    AVERAGE    AVERAGE    AVERAGE COMPUTATIONAL  AVERAGE    AVERAGE
PARAMETER      COUNT     CYCLE     NUMBER     QUANTUM   OVERHEAD EFFICIENCY    EXCHANGE   EXCHANGE
                         TIME     IN QUEUE                                     OVERHEAD     TIME

   1.0000  6168.0000    2.3308    16.7455      .0273      .8617    19.6359       .0051    1.0067

   2.0000  6089.0000    2.3613    16.6101      .0291      .8546    20.4897       .0050    1.0165

   3.0000  6270.0000    2.2970    17.0384      .0265      .8684    19.6698       .0051     .9766



  AVERAGE    MAXIMUM    MAXIMUM   MAXIMUM    MAXIMUM    MAXIMUM    MAXIMUM     STATIONS
 OVERLOAD    QUANTUM    CYCLE     NUMBER     OVERHEAD   OVERLOAD   NUMBER      SERVICED
                        TIME      IN QUEUE                         STATIONS

    .1369      .2000    3.5928    19.0000     1.0220     1.0000    20.0000     73.0000

    .1560      .2000    3.5928    19.0000     1.0120     1.0000    20.0000     78.0000

    .1657      .2000    3.5154    19.0000     1.0220     1.0000    20.0000     64.0000
```

FIGURE 3.2.7

SIMUALTOR OUTPUT DATA
WITH EARLY TERMINATIONS ALLOWED,
NOT ALLOWED AND BACKGROUND JOBS PERMITTED.
QUANTUM DETERMINATION METHOD Q_3, LARGE JOB DATA, AND 20 STATIONS

VARIABLE PARAMETER	CYCLE COUNT	AVERAGE CYCLE TIME	AVERAGE NUMBER IN QUEUE	AVERAGE QUANTUM	AVERAGE OVERHEAD	COMPUTATIONAL EFFICIENCY	AVERAGE EXCHANGE OVERHEAD	AVERAGE EXCHANGE TIME
1.0000	4520.0000	3.1804	13.2051	.1239	.6975	51.4356	.0040	.8039
2.0000	3940.0000	3.6490	12.5343	.1576	.6951	54.1443	.0041	.8049
3.0000	4674.0000	3.0816	13.3181	.1221	.6904	52.7852	.0040	.7516

AVERAGE OVERLOAD	MAXIMUM QUANTUM	MAXIMUM CYCLE TIME	MAXIMUM NUMBER IN QUEUE	MAXIMUM OVERHEAD	MAXIMUM OVERLOAD	MAXIMUM NUMBER STATIONS	STATIONS SERVICED
.0027	2.0000	4.8041	19.0000	1.0220	1.0000	20.0000	213.0000
.0028	2.0000	5.1038	19.0000	.9620	1.0000	20.0000	235.0000
.0015	2.0000	4.9651	19.0000	.9820	1.0000	20.0000	205.0000

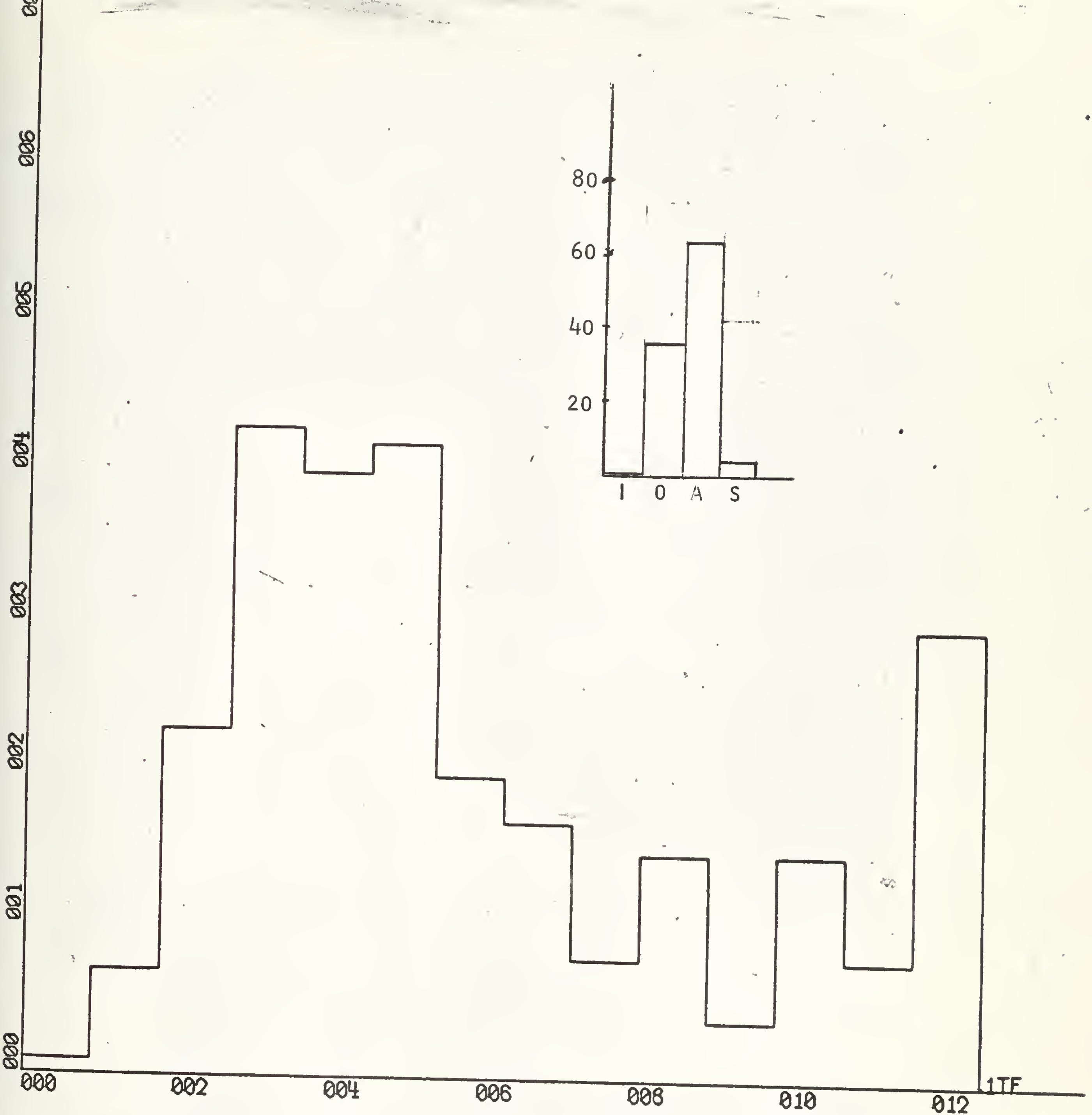
FIGURE 3.2.8

```
                         SIMUALTOR OUTPUT DATA
                   WITH EARLY TERMINATIONS ALLOWED,
              NOT ALLOWED AND BACKGROUND JOBS PERMITTED.
  QUANTUM DETERMINATION METHOD Q4, LARGE JOB DATA, AND 20 STATIONS

 VARIABLE     CYCLE   AVERAGE   AVERAGE   AVERAGE   AVERAGE COMPUTATIONAL AVERAGE   AVERAGE
PARAMETER     COUNT   CYCLE     NUMBER    QUANTUM  OVERHEAD EFFICIENCY    EXCHANGE  EXCHANGE
                      TIME     IN QUEUE                                   OVERHEAD   TIME

   1.0000 4429.0000   3.2458   12.4954     .1429     .6641   55.0165        .0038     .7447

   2.0000 3819.0000   3.7645   11.3501     .1915     .6395   57.7400        .0037     .7307

   3.0000 4624.0000   3.1145   12.3787     .1418     .6447   56.3414        .0037     .7001


 AVERAGE    MAXIMUM   MAXIMUM   MAXIMUM   MAXIMUM   MAXIMUM   MAXIMUM   STATIONS
 OVERLOAD   QUANTUM   CYCLE     NUMBER   OVERHEAD  OVERLOAD   NUMBER    SERVICED
                      TIME     IN QUEUE                       STATIONS

    .0023     .2000   6.0781   19.0000     .9720    1.0000   20.0000  229.0000

    .0029     .2000   5.9394   18.0000     .9120    1.0000   20.0000  259.0000

    .0041     .2000   6.2404   19.0000     .9820    1.0000   20.0000  224.0000
```

FIGURE 3.2.9

QUANTUM DETERMINATION METHOD Q_2, SCHOOL DATA, 20 STATIONS

WITH NO QUANTUM REDETERMINATION FOR EARLY COMPLETION POLICY



The step interval for abscissa for this and all other graphs in this section is 90 seconds.

X-SCALE = 2.00E+02 UNITS/INCH.
Y-SCALE = 1.00E+01 UNITS/INCH.

HISTOGRAM OF TOTAL TIME IN SYSTEM
OTTO/GRIMES THESIS

FIGURE 3.2.10

QUANTUM DETERMINATION METHOD Q_2, LARGE JOB DATA,

20 STATIONS WITH NO QUANTUM REDETERMINATION FOR EARLY COMPLETION POLICY

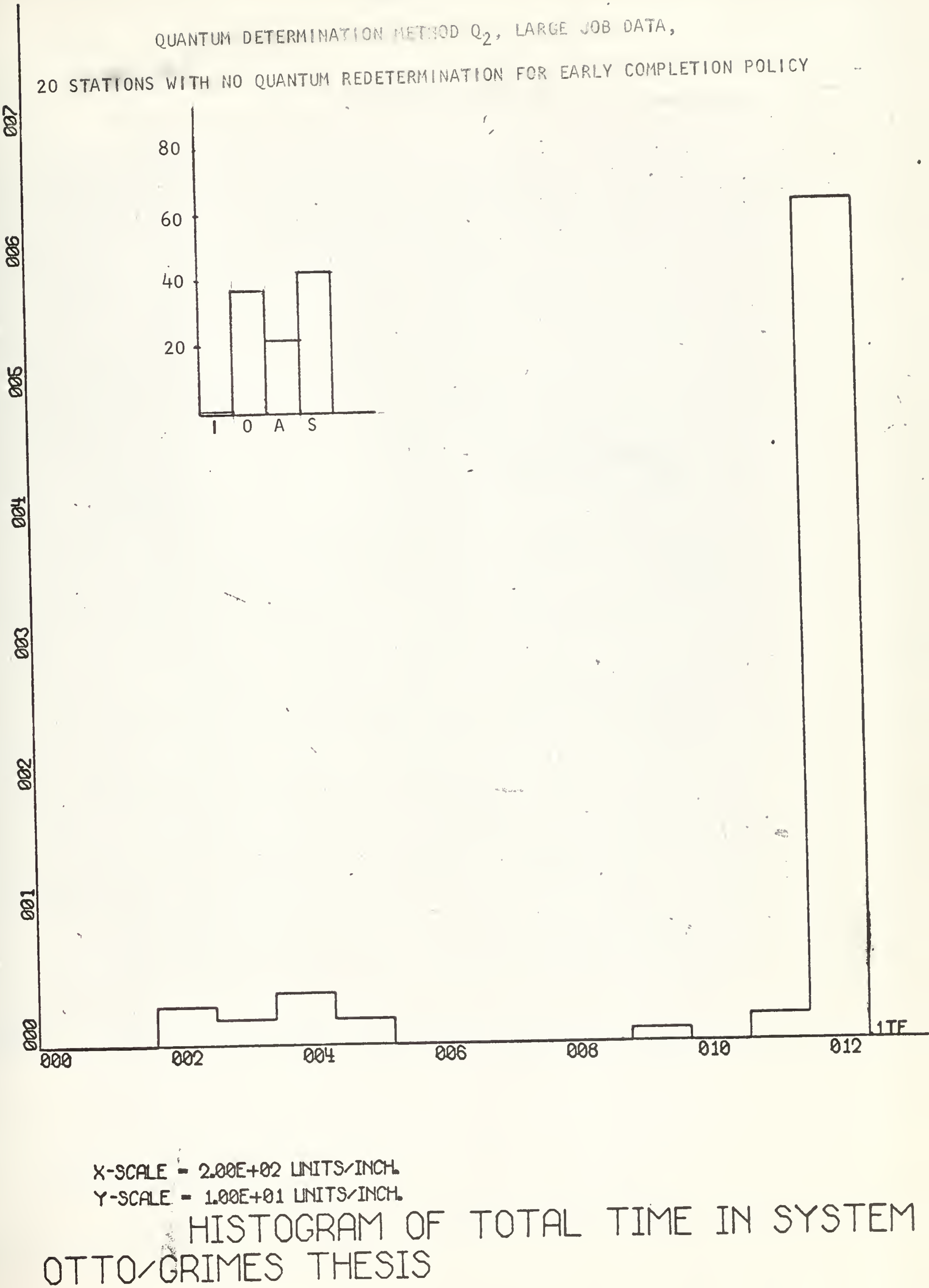
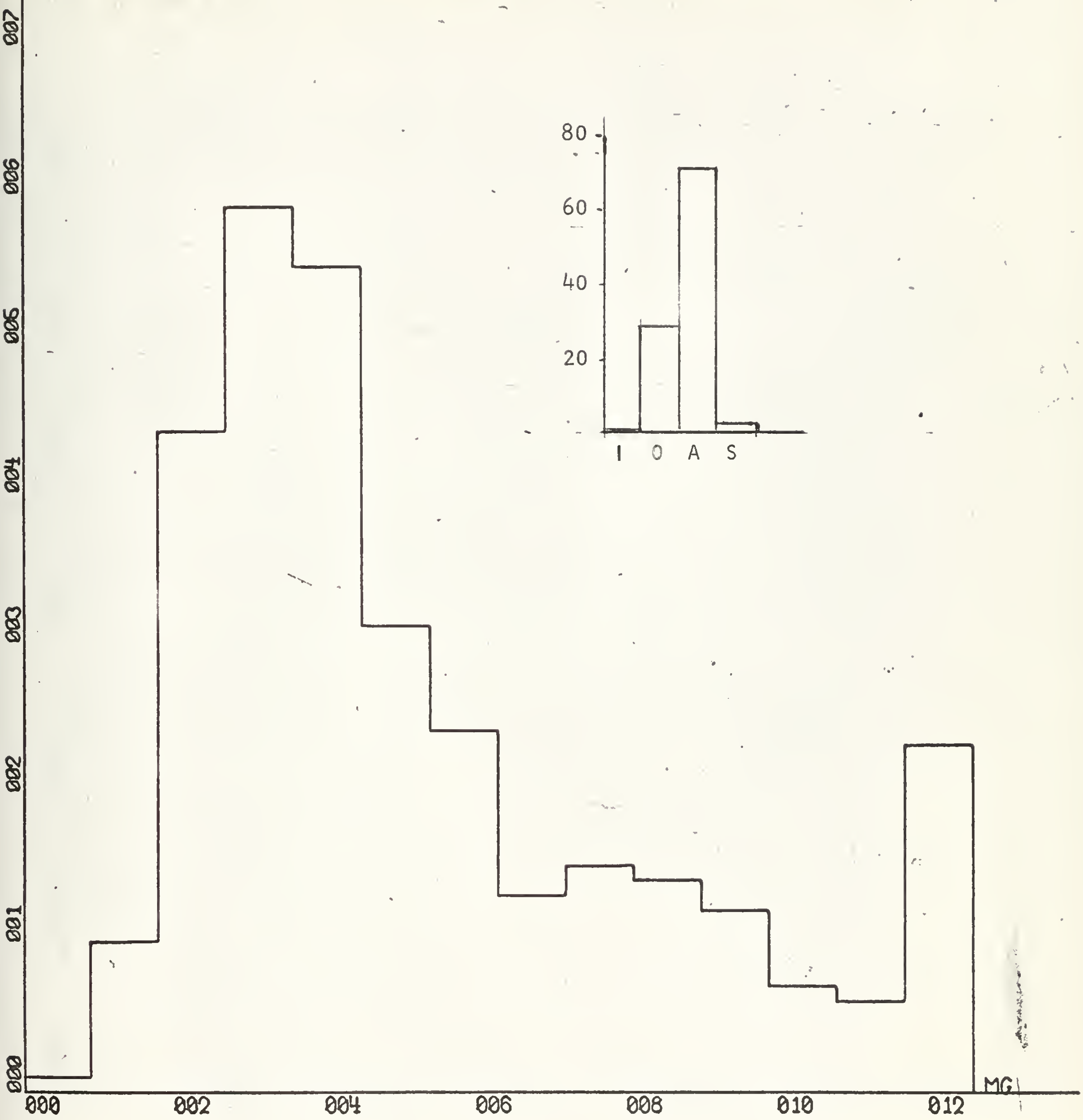


FIGURE 3.2.11

QUANTUM DETERMINATION METHOD Q_1, SCHOOL DATA, AND 20 STATIONS



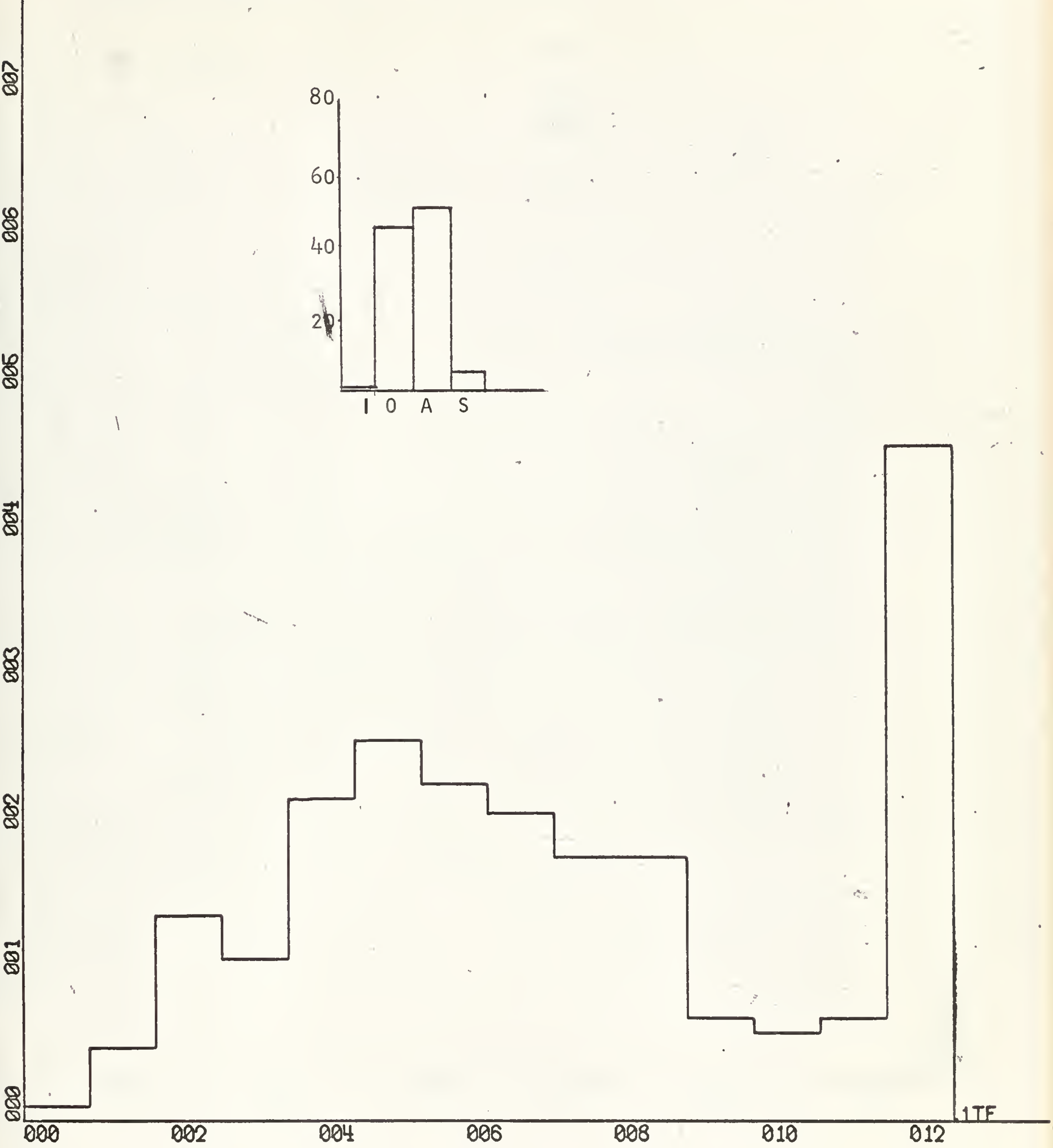
X-SCALE = 2.00E+02 UNITS/INCH.
Y-SCALE = 1.00E+01 UNITS/INCH.

HISTOGRAM OF TOTAL TIME IN SYSTEM

OTTO/GRIMES THESIS

FIGURE 3.2.12

QUANTUM DETERMINATION METHOD Q_2, SCHOOL DATA, AND 20 STATIONS



X-SCALE = 2.00E+02 UNITS/INCH.
Y-SCALE = 1.00E+01 UNITS/INCH.

HISTOGRAM OF TOTAL TIME IN SYSTEM
OTTO/GRIMES THESIS

FIGURE 3.2.13

QUANTUM DETERMINATION METHOD Q_3, SCHOOL DATA, AND 20 STATIONS

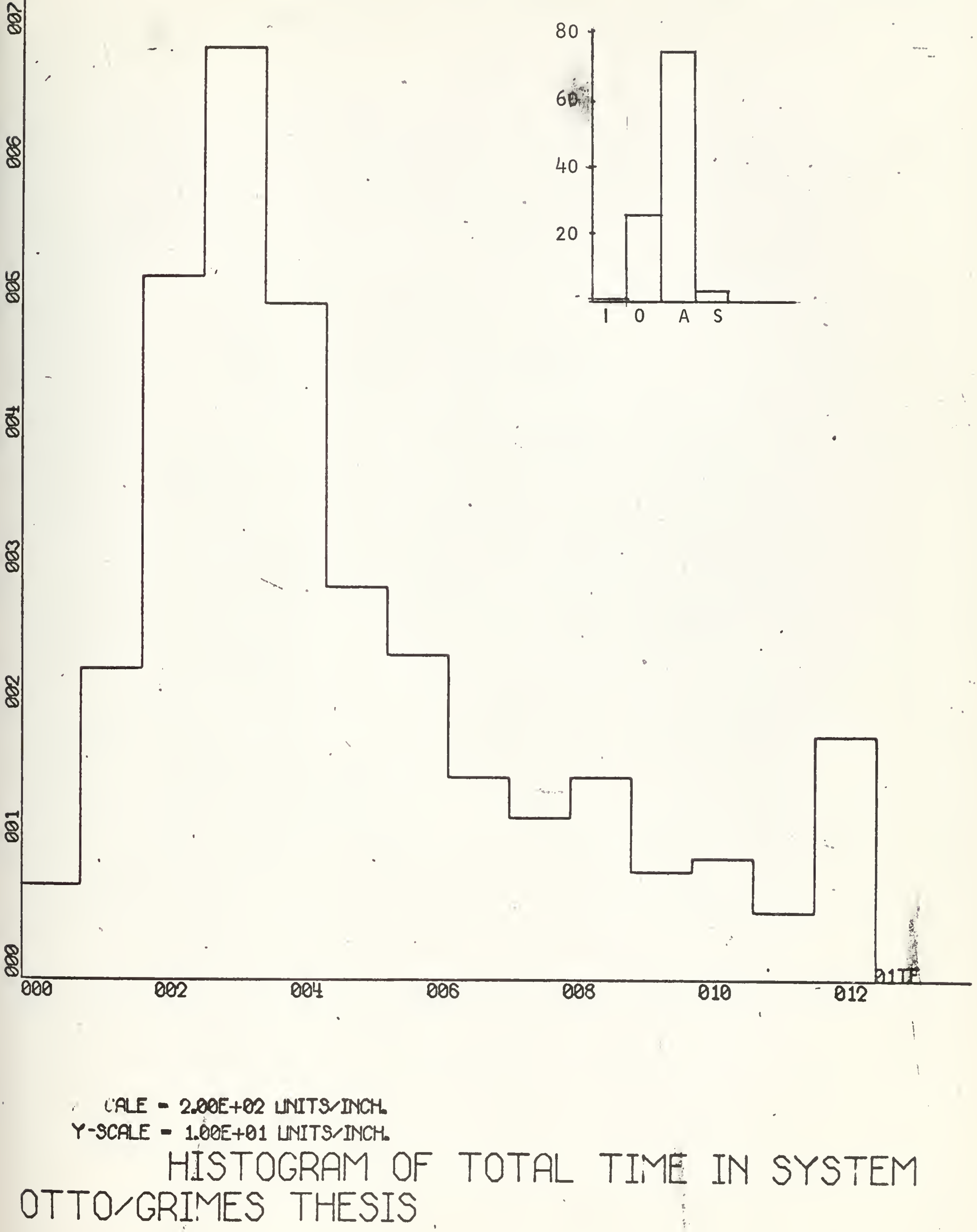
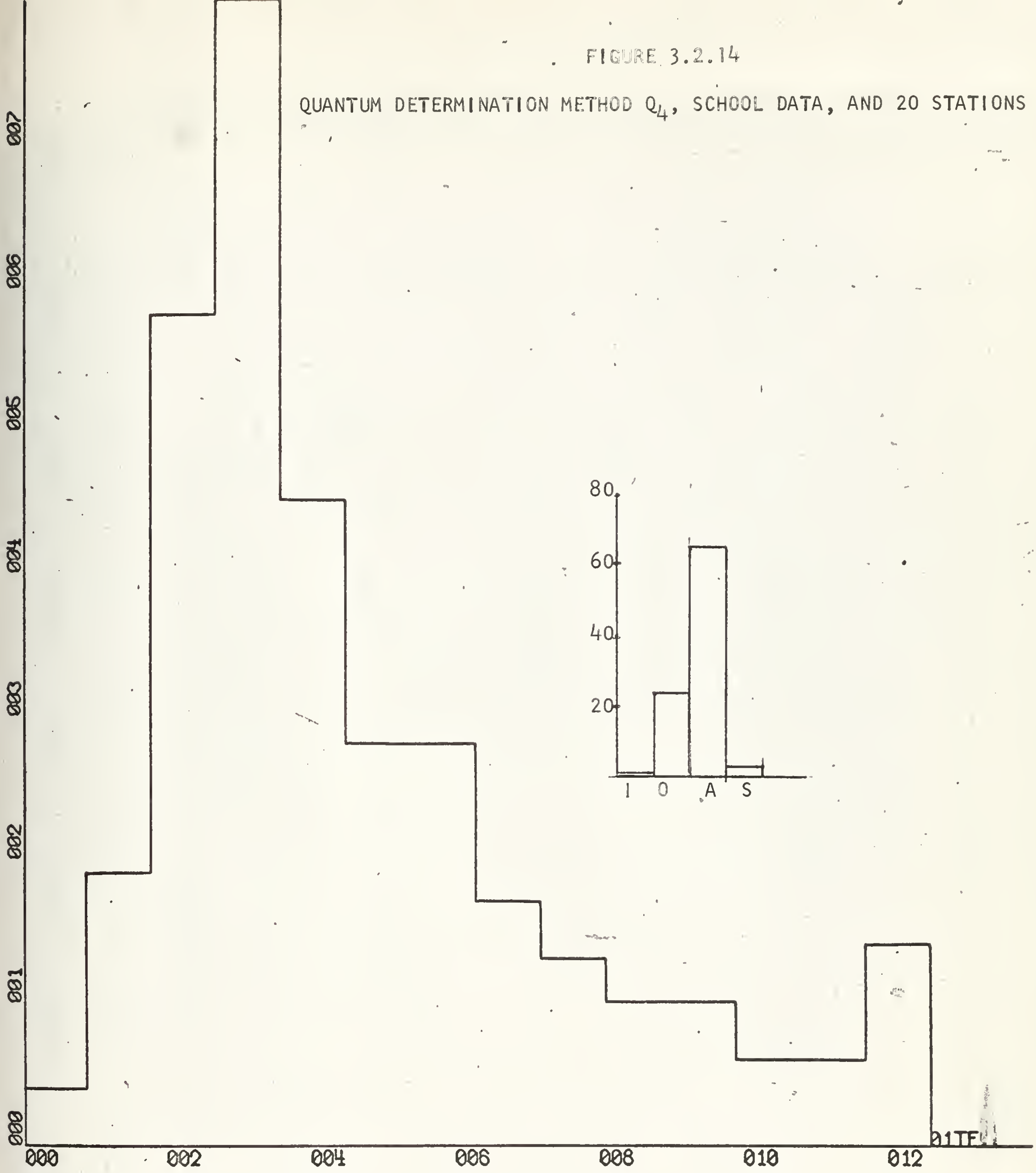


FIGURE 3.2.14

QUANTUM DETERMINATION METHOD Q_4, SCHOOL DATA, AND 20 STATIONS

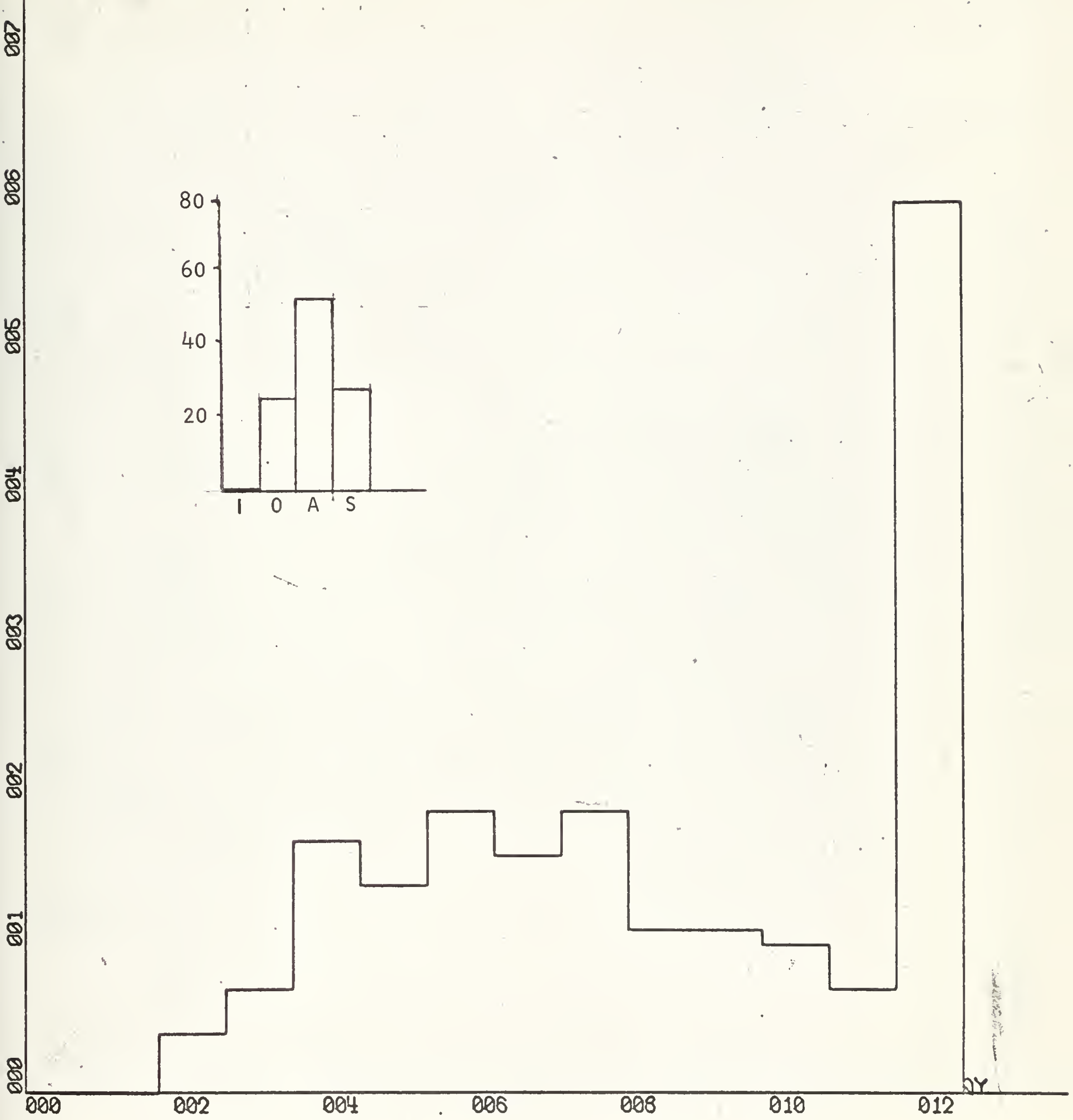


X-SCALE = 2.00E+02 UNITS/INCH.
Y-SCALE = 1.00E+01 UNITS/INCH.

HISTOGRAM OF TOTAL TIME IN SYSTEM
OTTO/GRIMES THESIS

FIGURE 3.2.15

QUANTUM DETERMINATION METHOD Q_1, LARGE JOB DATA, AND 20 STATIONS



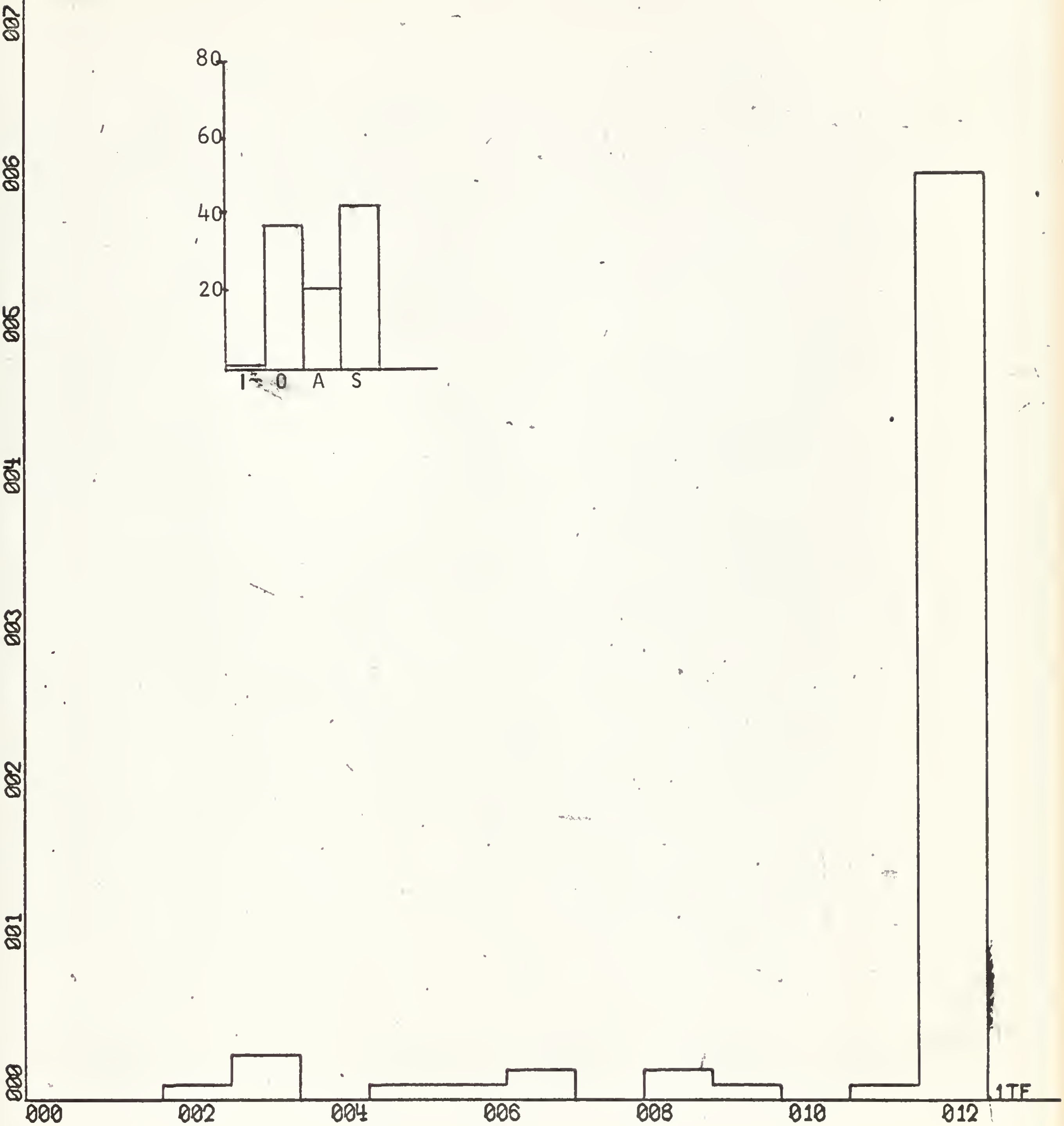
X-SCALE = 2.00E+02 UNITS/INCH.
Y-SCALE = 1.00E+01 UNITS/INCH.

HISTOGRAM OF TOTAL TIME IN SYSTEM

OTTO/GRIMES THESIS

FIGURE 3.2.16

QUANTUM DETERMINATION METHOD Q_2, LARGE JOB DATA, AND 20 STATIONS



X-SCALE = 2.00E+02 UNITS/INCH.
Y-SCALE = 1.00E+01 UNITS/INCH.

HISTOGRAM OF TOTAL TIME IN SYSTEM
OTTO/GRIMES THESIS

FIGURE 5.2.17

QUANTUM DETERMINATION METHOD Q_3, LARGE JOB DATA, AND 20 STATIONS

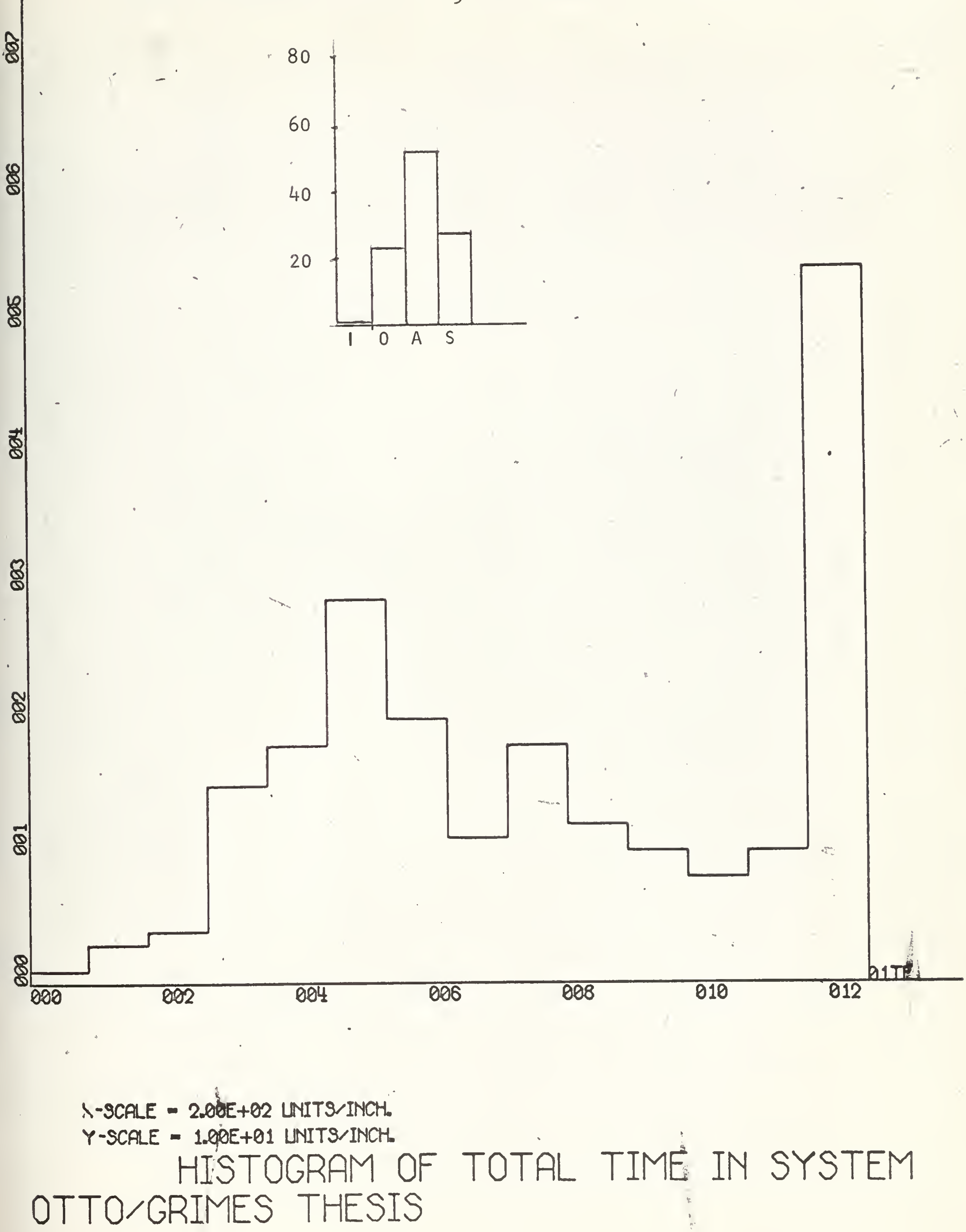
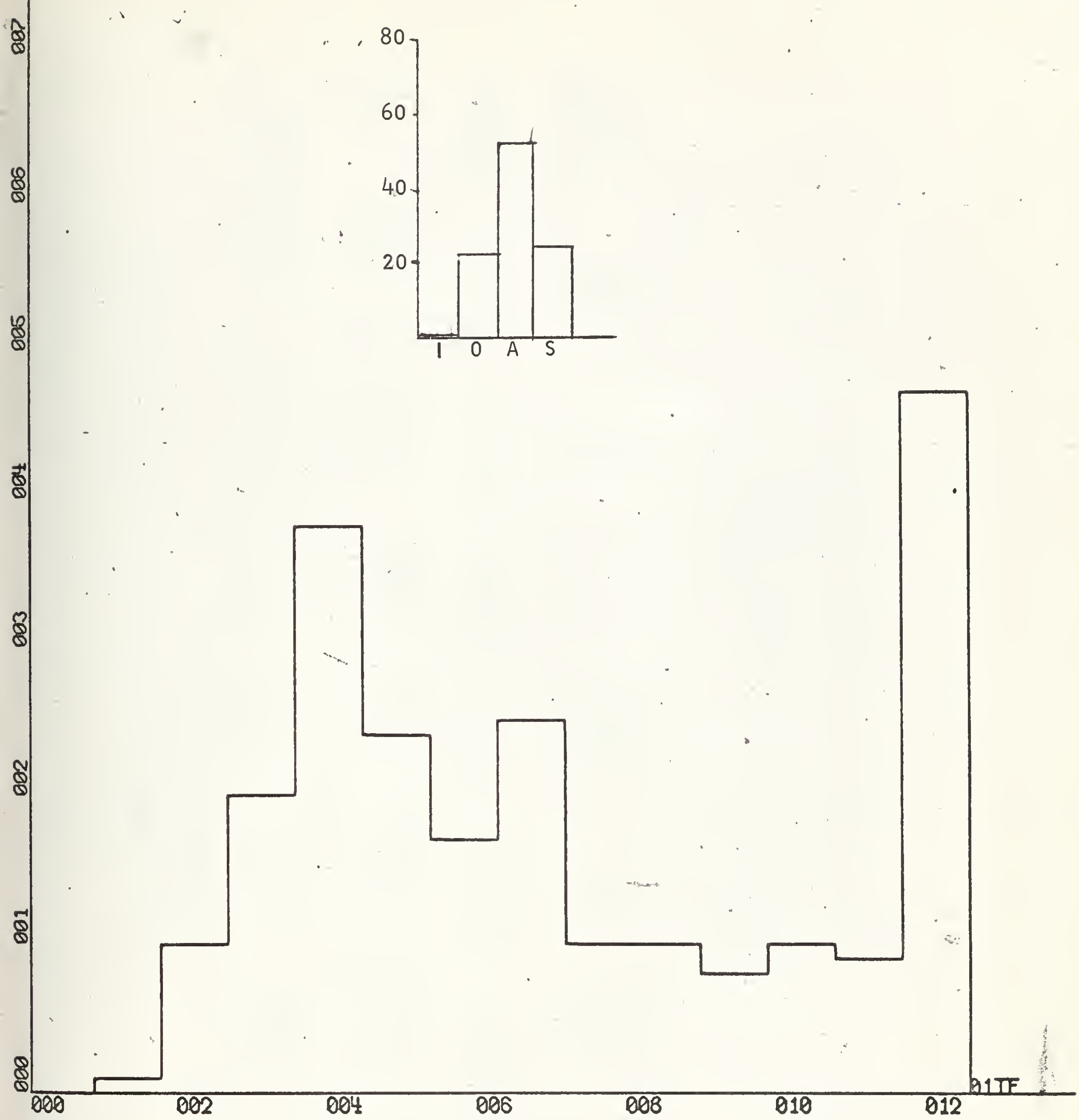


FIGURE 3.2.18

QUANTUM DETERMINATION METHOD Q_4, LARGE JOB DATA, AND 20 STATIONS



X-SCALE = 2.00E+02 UNITS/INCH.
Y-SCALE = 1.00E+01 UNITS/INCH.

HISTOGRAM OF TOTAL TIME IN SYSTEM

OTTO/GRIMES THESIS

FIGURE 3.3.1

SIMULATOR OUTPUT DATA

```
VARIABLE    CYCLE    AVERAGE   AVERAGE   AVERAGE   AVERAGE  COMPUTATIONAL .AVERAGE  AVERAGE
PARAMETER   COUNT    CYCLE     NUMBER    QUANTUM   OVERHEAD EFFICIENCY     EXCHANGE EXCHANGE
                     TIME      IN QUEUE                                    OVERHEAD TIME

    2       4900     2.9342    13.1131    .105     .7030       48.70       .0041     .7910
    4       3947     3.6425    11.2057    .188     .6366       62.41       .0036     .7115
    6       3906     3.6814    11.0929    .190     .6387       62.63       .0036     .7156
    8       3906     3.6814    11.0929    .190     .6387       62.63       .0036     .7156
   10       3906     3.6814    11.0929    .190     .6387       62.63       .0036     .7156

AVERAGE   MAXIMUM   MAXIMUM   MAXIMUM   MAXIMUM   MAXIMUM   MAXIMUM   STATIONS   AVERAGE
OVERLOAD  QUANTUM   CYCLE     NUMBER    OVERHEAD  OVERLOAD  NUMBER    SERVICED   SERVICE
                    TIME      IN QUEUE                      STATIONS             TIME

  .0014    .2000    4.5362      20      1.0220      1         20        193     1001.22
  .0053    .2000    5.9986      19       .9920      1         20        256      724.09
  .0033    .2000    6.2005      19       .9920      1         20        256      716.77
  .0033    .2000    6.2005      19       .9920      1         20        256      716.77
  .0033    .2000    6.2005      19       .9920      1         20        256      716.77
```

LARGE JOB DATA   POLICY NO. 1
QUANTUM DETERMINATION METHOD ONE
VARIABLE PARAMETER - RESPONSE CYCLE TIME

FIGURE 3.3.2

SIMULATOR OUTPUT DATA

```
VARIABLE     CYCLE    AVERAGE   AVERAGE    AVERAGE  AVERAGE  COMPUTATIONAL AVERAGE  AVERAGE
PARAMETER    COUNT    CYCLE     NUMBER     QUANTUM  OVERHEAD EFFICIENCY    EXCHANGE EXCHANGE
                       TIME     IN QUEUE                                   OVERHEAD TIME

    2        4159    3.4571    19.1479     .068    1.0081       38.86      .0059   1.0968
    4        2921    4.9232    18.5892     .140    1.0142       55.87      .0059   1.1333
    6        2413    5.9594    18.0406     .190    1.0285       62.71      .0059   1.1576
    8        2416    5.9512    18.0389     .190    1.0267       62.71      .0059   1.1564
   10        2416    5.9512    18.0389     .190    1.0267       62.71      .0059   1.1564

AVERAGE     MAXIMUM  MAXIMUM  MAXIMUM   MAXIMUM   MAXIMUM   MAXIMUM   STATIONS    AVERAGE
OVERLOAD    QUANTUM  CYCLE    NUMBER    OVERHEAD  OVERLOAD  NUMBER    SERVICED    SERVICE
                     TIME     IN QUEUE                      STATIONS              TIME

 8.4720      .2000   4.7277     20      1.1320     10         30        149      1997.41
 6.7467      .2000   6.5614     20      1.1520      9         30        227      1375.93
 5.8214      .2000   7.5616     20      1.1620      9         30        256      1205.07
 5.8137      .2000   7.5616     20      1.1620      9         30        255      1199.24
 5.8137      .2000   7.5616     20      1.1620      9         30        255      1199.24
```

LARGE JOB DATA   POLICY NO. 1
QUANTUM DETERMINATION METHOD ONE
VARIABLE PARAMETER - RESPONSE CYCLE TIME

```
                                              FIGURE 3.3.3

                                         SIMULATOR OUTPUT DATA

VARIABLE       CYCLE        AVERAGE    AVERAGE    AVERAGE   AVERAGE  COMPUTATIONAL AVERAGE   AVERAGE
PARAMETER      COUNT         CYCLE     NUMBER     QUANTUM   OVERHEAD EFFICIENCY    EXCHANGE EXCHANGE
                             TIME      IN QUEUE                                    OVERHEAD TIME

    2          3539         4.0626     24.1568     .056     1.2565       33.93        .0074    1.4200
    4          2739         5.2503     21.3611     .124     1.1524       52.59        .0067    1.3127
    6          2316         6.2092     19.7427     .183     1.0960       61.89        .0063    1.2338
    8          2265         6.3496     19.5254     .190     1.0943       62.70        .0063    1.2359
   10          2243         6.4119     19.5025     .190     1.1046       62.70        .0064    1.2485

AVERAGE      MAXIMUM      MAXIMUM    MAXIMUM     MAXIMUM    MAXIMUM    MAXIMUM    STATIONS     AVERAGE
OVERLOAD     QUANTUM      CYCLE      NUMBER      OVERHEAD   OVERLOAD   NUMBER     SERVICED     SERVICE
                          TIME       IN QUEUE                          STATIONS                TIME

  .0020        .2000      5.5291       30         1.5420        1         30         131        2597.18
  .0088        .2000      7.2725       29         1.4920        1         30         212        1519.82
  .0194        .2000      8.4760       28         1.4520        1         30         251        1232.76
  .0216        .2000      8.8259       28         1.4420        1         30         255        1225.11
  .0156        .2000      8.8365       28         1.4620        1         30         254        1226.43


                                 LARGE JOB DATA              POLICY NO. 1
                                    QUANTUM DETERMINATION METHOD ONE
                                 VARIABLE PARAMETER - RESPONSE CYCLE TIME
```

FIGURE 3.3.4

SIMULATOR OUTPUT DATA

```
VARIABLE       CYCLE     AVERAGE    AVERAGE     AVERAGE   AVERAGE   COMPUTATIONAL  AVERAGE   AVERAGE
PARAMETER      COUNT      CYCLE     NUMBER      QUANTUM   OVERHEAD  EFFICIENCY     EXCHANGE  EXCHANGE
                          TIME      IN QUEUE                                       OVERHEAD  TIME

    2          3246      4.4328     28.2693      .046     1.4624       29.63        .0086    1.6509
    4          2592      5.5513     26.5868      .093     1.4056       45.88        .0082    1.5776
    6          2062      6.9812     24.5548      .159     1.3318       58.50        .0077    1.5270
    8          1892      7.6071     23.6897      .188     1.3158       62.42        .0076    1.4971
   10          1857      7.7500     23.6198      .190     1.3326       62.68        .0077    1.5127

AVERAGE      MAXIMUM    MAXIMUM    MAXIMUM     MAXIMUM    MAXIMUM   MAXIMUM     STATIONS   AVERAGE
OVERLOAD     QUANTUM    CYCLE      NUMBER      OVERHEAD   OVERLOAD  NUMBER      SERVICED   SERVICE
                        TIME       IN QUEUE                         STATIONS               TIME

 2.7356       .2000     5.8419       30        1.6120        5         35          106     3380.54
  .9726       .2000     8.3574       30        1.6420        4         35          181     2134.69
  .2774       .2000     9.5899       30        1.6120        3         35          234     1587.06
  .1152       .2000     9.5976       30        1.6220        2         35          250     1467.06
  .1686       .2000     9.9469       30        1.6520        3         35          251     1457.88
```

LARGE JOB DATA   POLICY NO. 1
QUANTUM DETERMINATION METHOD ONE
VARIABLE PARAMETER - RESPONSE CYCLE TIME

FIGURE 3.3.5

SIMULATOR OUTPUT DATA

VARIABLE PARAMETER	CYCLE COUNT	AVERAGE CYCLE TIME	AVERAGE NUMBER IN QUEUE	AVERAGE QUANTUM	AVERAGE OVERHEAD	COMPUTATIONAL EFFICIENCY	AVERAGE EXCHANGE OVERHEAD	AVERAGE EXCHANGE TIME
2	3120	4.6128	29.7433	.045	1.5331	29.51	.0091	1.7124
4	2596	5.5435	27.0054	.092	1.4138	45.73	.0083	1.5743
6	2078	6.9262	24.6689	.158	1.3287	58.51	.0077	1.5062
8	1873	7.6826	23.6962	.188	1.3290	62.46	.0077	1.5085
10	1880	7.6570	23.6436	.190	1.3163	62.67	.0076	1.4954

AVERAGE OVERLOAD	MAXIMUM QUANTUM	MAXIMUM CYCLE TIME	MAXIMUM NUMBER IN QUEUE	MAXIMUM OVERHEAD	MAXIMUM OVERLOAD	MAXIMUM NUMBER STATIONS	STATIONS SERVICED	AVERAGE SERVICE TIME
.0026	.2000	6.0867	35	1.7920	1	35	105	3469.85
.0081	.2000	7.8521	34	1.7420	1	35	179	2111.25
.0294	.2000	9.3358	34	1.7420	2	35	232	1593.52
.0470	.2000	9.7418	33	1.7020	2	35	251	1464.19
.0511	.2000	10.2201	33	1.7120	2	35	252	1458.52

LARGE JOB DATA POLICY NO. 1
QUANTUM DETERMINATION METHOD ONE
VARIABLE PARAMETER - RESPONSE CYCLE TIME

FIGURE 3.3.6

SIMULATOR OUTPUT DATA

VARIABLE PARAMETER	CYCLE COUNT	AVERAGE CYCLE TIME	AVERAGE NUMBER IN QUEUE	AVERAGE QUANTUM	AVERAGE OVERHEAD	COMPUTATIONAL EFFICIENCY	AVERAGE EXCHANGE OVERHEAD	AVERAGE EXCHANGE TIME
6	5223	5.5070	10.1767	.353	.6345	75.53	.0035	.6742
8	5197	5.5341	10.1380	.361	.6275	75.89	.0035	.6677
10	5193	5.5382	10.1080	.361	.6280	75.90	.0035	.6678

AVERAGE OVERLOAD	MAXIMUM QUANTUM	MAXIMUM CYCLE TIME	MAXIMUM NUMBER IN QUEUE	MAXIMUM OVERHEAD	MAXIMUM OVERLOAD	MAXIMUM NUMBER STATIONS	STATIONS SERVICED	AVERAGE SERVICE TIME
.0172	.4000	8.0990	18	.9420	1	20	615	590.41
.0165	.4000	9.2255	18	.9820	1	20	616	591.10
.0258	.4000	8.9019	18	.9620	1	20	617	588.14

LARGE JOB DATA POLICY NO. 1
QUANTUM DETERMINATION METHOD ONE
VARIABLE PARAMETER - RESPONSE CYCLE TIME

FIGURE 3.4.1

SIMUALTOR OUTPUT DATA
WITH EARLY TERMINATIONS ALLOWED,
NOT ALLOWED AND BACKGROUND JOBS PERMITTED
QUANTUM DETERMINATION METHOD Q_1, SCHOOL DATA, AND 10 STATIONS

VARIABLE PARAMETER	CYCLE COUNT	AVERAGE CYCLE TIME	AVERAGE NUMBER IN QUEUE	AVERAGE QUANTUM	AVERAGE OVERHEAD	COMPUTATIONAL EFFICIENCY	AVERAGE EXCHANGE OVERHEAD	AVERAGE EXCHANGE TIME	AVERAGE OVERLOAD	MAXIMUM QUANTUM	MAXIMUM CYCLE TIME	MAXIMUM NUMBER IN QUEUE	MAXIMUM OVERHEAD	MAXIMUM OVERLOAD	MAXIMUM NUMBER STATIONS	STATIONS SERVICED
1.00	20608.0000	.5995	2.2141	.1916	.1331	70.7487	.0007	.0148	.0002	.2000	2.3267	9.0000	.4920	1.0000	10.0000	295.0000
2.00	20752.0000	.5942	2.2033	.1916	.1311	71.0400	.0007	.0148	.0002	.2000	2.0959	9.0000	.4620	1.0000	10.0000	295.0000
3.00	18805.0000	.7658	2.9956	.1944	.1647	76.0531	.0009	.0177	0	.2000	2.2963	9.0000	.4720	0	10.0000	232.0000

FIGURE 3.4.2

SIMUALTOR OUTPUT DATA
WITH EARLY TERMINATIONS ALLOWED,
NOT ALLOWED AND BACKGROUND JOBS PERMITTED
QUANTUM DETERMINATION METHOD Q_4, SCHOOL DATA, AND 10 STATIONS

VARIABLE PARAMETER	CYCLE COUNT	AVERAGE CYCLE TIME	AVERAGE NUMBER IN QUEUE	AVERAGE QUANTUM	AVERAGE OVERHEAD	COMPUTATIONAL EFFICIENCY	AVERAGE EXCHANGE OVERHEAD	AVERAGE EXCHANGE TIME
1.0000	5200.0000	2.7651	18.0740	.0897	.9536	58.6631	.0056	.1411
2.0000	4847.0000	2.9660	18.0671	.0989	.9583	60.2543	.0056	.1448
3.0000	5206.0000	2.7663	18.1028	.0894	.9542	58.5076	.0056	.1467

AVERAGE OVERLOAD	MAXIMUM QUANTUM	MAXIMUM CYCLE TIME	MAXIMUM NUMBER IN QUEUE	MAXIMUM OVERHEAD	MAXIMUM OVERLOAD	MAXIMUM NUMBER STATIONS	STATIONS SERVICED
6.3212	.2000	3.3284	19.0000	1.0820	11.0000	30.0000	269.0000
3.6036	.2000	3.3380	19.0000	1.0620	11.0000	30.0000	280.0000
6.2196	.2000	3.3297	19.0000	1.0820	11.0000	30.0000	262.0000

FIGURE 3.4.3

SIMUALTOR OUTPUT DATA
WITH EARLY TERMINATIONS ALLOWED,
NOT ALLOWED AND BACKGROUND JOBS PERMITTED
QUANTUM DETERMINATION METHOD Q_1, LARGE JOB DATA, AND 10 STATIONS

VARIABLE PARAMETER	CYCLE COUNT	AVERAGE CYCLE TIME	AVERAGE NUMBER IN QUEUE	AVERAGE QUANTUM	AVERAGE OVERHEAD	COMPUTATIONAL EFFICIENCY	AVERAGE EXCHANGE OVERHEAD	AVERAGE EXCHANGE TIME
1.0000	5117.0000	2.8124	18.4335	.0887	.9746	58.1619	.0057	.1490
2.0000	4765.0000	3.0201	18.3922	.0987	.9772	60.0924	.0058	.1474
3.0000	5126.0000	2.8097	18.4629	.0886	.9745	58.2115	.0057	.1492

AVERAGE OVERLOAD	MAXIMUM QUANTUM	MAXIMUM CYCLE TIME	MAXIMUM NUMBER IN QUEUE	MAXIMUM OVERHEAD	MAXIMUM OVERLOAD	MAXIMUM NUMBER STATIONS	STATIONS SERVICED
22.8423	.2000	3.3358	19.0000	1.0920	21.0000	40.0000	263.0000
13.3715	.2000	3.3158	19.0000	1.0620	21.0000	40.0000	280.0000
22.7563	.2000	3.3717	19.0000	1.1120	21.0000	40.0000	263.0000

FIGURE 3.4.4

SIMUALTOR OUTPUT DATA
WITH EARLY TERMINATIONS ALLOWED,
NOT ALLOWED AND BACKGROUND JOBS PERMITTED
QUANTUM DETERMINATION METHOD Q_4, LARGE JOB DATA, AND 10 STATIONS

VARIABLE PARAMETER	CYCLE COUNT	AVERAGE CYCLE TIME	AVERAGE NUMBER IN QUEUE	AVERAGE QUANTUM	AVERAGE OVERHEAD	COMPUTATIONAL EFFICIENCY	AVERAGE EXCHANGE OVERHEAD	AVERAGE EXCHANGE TIME
1.0000	5030.0000	2.8614	18.8006	.0884	.9937	58.0706	.0058	.1519
2.0000	4673.0000	3.0801	18.7860	.0981	1.0006	59.8532	.0059	.1483
3.0000	5083.0000	2.8331	18.7199	.0881	.9866	58.2435	.0058	.1478

AVERAGE OVERLOAD	MAXIMUM QUANTUM	MAXIMUM CYCLE TIME	MAXIMUM NUMBER IN QUEUE	MAXIMUM OVERHEAD	MAXIMUM OVERLOAD	MAXIMUM NUMBER STATIONS	STATIONS SERVICED
39.9022	.2000	3.3405	19.0000	1.1220	30.0000	50.0000	260.0000
23.7393	.2000	3.3607	19.0000	1.1120	31.0000	50.0000	276.0000
39.5786	.2000	3.3644	19.0000	1.0820	30.0000	50.0000	262.0000

FIGURE 3.4.5

SIMUALTOR OUTPUT DATA
WITH EARLY TERMINATIONS ALLOWED,
NOT ALLOWED AND BACKGROUND JOBS PERMITTED
QUANTUM DETERMINATION METHOD Q_1, SCHOOL DATA, AND 30 STATIONS

VARIABLE PARAMETER	CYCLE COUNT	AVERAGE CYCLE TIME	AVERAGE NUMBER IN QUEUE	AVERAGE QUANTUM	AVERAGE OVERHEAD	COMPUTATIONAL EFFICIENCY	AVERAGE EXCHANGE OVERHEAD	AVERAGE EXCHANGE TIME
1.00	15717.0000	.8282	2.5918	.1898	.1558	59.4076	.0003	.1392
2.00	15340.0000	.8475	2.6493	.1899	.1574	59.3537	.0009	.1440
3.00	13317.0000	1.0814	3.6544	.1928	.1987	65.1669	.0011	.1740

AVERAGE OVERLOAD	MAXIMUM QUANTUM	MAXIMUM CYCLE TIME	MAXIMUM NUMBER IN QUEUE	MAXIMUM OVERHEAD	MAXIMUM OVERLOAD	MAXIMUM NUMBER STATIONS	STATIONS SERVICED
.0001	.2000	2.9593	9.0000	.4820	1.0000	10.0000	237.0000
0	.2000	2.9593	9.0000	.4620	0	10.0000	237.0000
0	.2000	2.9649	8.0000	.4420	0	10.0000	189.0000

FIGURE 3.4.6

SIMUALTOR OUTPUT DATA
WITH EARLY TERMINATIONS ALLOWED,
NOT ALLOWED AND BACKGROUND JOBS PERMITTED
QUANTUM DETERMINATION METHOD Q_4, SCHOOL DATA, AND 30 STATIONS

VARIABLE PARAMETER	CYCLE COUNT	AVERAGE CYCLE TIME	AVERAGE NUMBER IN QUEUE	AVERAGE QUANTUM	AVERAGE OVERHEAD	COMPUTATIONAL EFFICIENCY	AVERAGE EXCHANGE OVERHEAD	AVERAGE EXCHANGE TIME
1.0000	4054.0000	3.5471	18.1934	.0821	.9612	42.1194	.0056	1.0365
2.0000	3651.0000	3.9382	18.0498	.0985	.9655	45.1465	.0057	1.0909
3.0000	4074.0000	3.5349	18.0668	.0829	.9548	42.3738	.0056	1.0264

AVERAGE OVERLOAD	MAXIMUM QUANTUM	MAXIMUM CYCLE TIME	MAXIMUM NUMBER IN QUEUE	MAXIMUM OVERHEAD	MAXIMUM OVERLOAD	MAXIMUM NUMBER STATIONS	STATIONS SERVICED
9.4930	.2000	5.0830	19.0000	1.0820	11.0000	30.0000	171.0000
5.0814	.2000	5.5360	19.0000	1.0620	11.0000	30.0000	187.0000
9.6413	.2000	5.1540	19.0000	1.0920	11.0000	30.0000	169.0000

FIGURE 3.4.7

SIMUALTOR OUTPUT DATA
WITH EARLY TERMINATIONS ALLOWED,
NOT ALLOWED AND BACKGROUND JOBS PERMITTED
QUANTUM DETERMINATION METHOD Q_1, LARGE JOB DATA, AND 30 STATIONS

VARIABLE PARAMETER	CYCLE COUNT	AVERAGE CYCLE TIME	AVERAGE NUMBER IN QUEUE	AVERAGE QUANTUM	AVERAGE OVERHEAD	COMPUTATIONAL EFFICIENCY	AVERAGE EXCHANGE OVERHEAD	AVERAGE EXCHANGE TIME
1.0000	3966.0000	3.6285	18.4498	.0825	.9781	41.9621	.0057	1.0682
2.0000	3627.0000	3.9681	18.4014	.0983	.9787	45.6009	.0057	1.0966
3.0000	4022.0000	3.5811	18.4170	.0822	.9705	42.2564	.0057	1.0482

AVERAGE OVERLOAD	MAXIMUM QUANTUM	MAXIMUM CYCLE TIME	MAXIMUM NUMBER IN QUEUE	MAXIMUM OVERHEAD	MAXIMUM OVERLOAD	MAXIMUM NUMBER STATIONS	STATIONS SERVICED
26.9173	.2000	5.1079	19.0000	1.0820	20.0000	40.0000	164.0000
14.8185	.2000	5.3199	19.0000	1.0620	20.0000	40.0000	183.0000
26.7583	.2000	4.9622	19.0000	1.1420	20.0000	40.0000	167.0000

FIGURE 3.4.8

SIMUALTOR OUTPUT DATA
WITH EARLY TERMINATIONS ALLOWED,
NOT ALLOWED AND BACKGROUND JOBS PERMITTED
QUANTUM DETERMINATION METHOD Q_4, LARGE JOB DATA, AND 30 STATIONS

VARIABLE PARAMETER	CYCLE COUNT	AVERAGE CYCLE TIME	AVERAGE NUMBER IN QUEUE	AVERAGE QUANTUM	AVERAGE OVERHEAD	COMPUTATIONAL EFFICIENCY	AVERAGE EXCHANGE OVERHEAD	AVERAGE EXCHANGE TIME
1.0000	3960.0000	3.6347	18.8129	.0820	1.0070	42.4624	.0057	1.0586
2.0000	3606.0000	3.9913	18.7965	.0978	.9970	46.0645	.0058	1.0961
3.0000	3960.0000	3.6183	18.6854	.0818	.9938	42.2489	.0057	1.0605

AVERAGE OVERLOAD	MAXIMUM QUANTUM	MAXIMUM CYCLE TIME	MAXIMUM NUMBER IN QUEUE	MAXIMUM OVERHEAD	MAXIMUM OVERLOAD	MAXIMUM NUMBER STATIONS	STATIONS SERVICED
43.5323	.2000	5.1720	19.0000	1.0920	29.0000	50.0000	164.0000
24.3337	.2000	5.2721	19.0000	1.1120	31.0000	50.0000	187.0000
43.6510	.2000	5.2138	19.0000	1.0920	30.0000	50.0000	163.0000

FIGURE 3.4.9

SIMUALTOR OUTPUT DATA
WITH EARLY TERMINATIONS ALLOWED, NOT ALLOWED AND BACKGROUND JOBS PERMITTED

QUANTUM DETERMINATION METHOD Q_1, SCHOOL DATA, AND 40 STATIONS

```
 VARIABLE       CYCLE   AVERAGE    AVERAGE    AVERAGE    AVERAGE COMPUTATIONAL AVERAGE   AVERAGE
PARAMETER       COUNT    CYCLE     NUMBER     QUANTUM   OVERHEAD EFFICIENCY    EXCHANGE  EXCHANGE
                         TIME     IN QUEUE                                     OVERHEAD   TIME

   1.0000 2699.0000     4.0883     3.0619      .8753      .2591   65.5508       .0013     .0308

   2.0000 2398.0000     4.6098     2.9458      .9721      .2623   62.1190       .0013     .0307

   3.0000 6726.0000     2.1410     3.3451      .5765      .1876   90.0688       .0010     .0204



 AVERAGE     MAXIMUM    MAXIMUM    MAXIMUM    MAXIMUM    MAXIMUM    MAXIMUM    STATIONS
OVERLOAD     QUANTUM     CYCLE     NUMBER    OVERHEAD   OVERLOAD    NUMBER     SERVICED
                         TIME     IN QUEUE                          STATIONS

    .0030     2.0000    6.8971     8.0000      .4320     1.0000    10.0000   326.0000

    .0025     2.0000    6.8025     7.0000      .3620     1.0000    10.0000   332.0000

    .0006     2.0000    2.9501     9.0000      .4820     1.0000    10.0000   239.0000
```

FIGURE 3.4.10

SIMUALTOR OUTPUT DATA
WITH EARLY TERMINATIONS ALLOWED,
NOT ALLOWED AND BACKGROUND JOBS PERMITTED
QUANTUM DETERMINATION METHOD Q_4, SCHOOL DATA, AND 40 STATIONS

VARIABLE PARAMETER	CYCLE COUNT	AVERAGE CYCLE TIME	AVERAGE NUMBER IN QUEUE	AVERAGE QUANTUM	AVERAGE OVERHEAD	COMPUTATIONAL EFFICIENCY	AVERAGE EXCHANGE OVERHEAD	AVERAGE EXCHANGE TIME
1.0000	5019.0000	2.8644	18.5878	.0917	.9751	59.5031	.0057	.1440
2.0000	4518.0000	3.1823	18.5142	.1064	.9832	61.9134	.0058	.1429
3.0000	5056.0000	2.8483	18.5473	.0915	.9691	59.6108	.0057	.1474

AVERAGE OVERLOAD	MAXIMUM QUANTUM	MAXIMUM CYCLE TIME	MAXIMUM NUMBER IN QUEUE	MAXIMUM OVERHEAD	MAXIMUM OVERLOAD	MAXIMUM NUMBER STATIONS	STATIONS SERVICED
6.6300	2.0000	3.4165	19.0000	1.1020	10.0000	30.0000	269.0000
3.3133	2.0000	3.5129	19.0000	1.0620	11.0000	30.0000	296.0000
6.2996	2.0000	3.5212	19.0000	1.1120	11.0000	30.0000	270.0000

FIGURE 3.4.11

SIMUALTOR OUTPUT DATA
WITH EARLY TERMINATIONS ALLOWED,
NOT ALLOWED AND BACKGROUND JOBS PERMITTED

QUANTUM DETERMINATION METHOD Q_1, LARGE JOB DATA, AND 40 STATIONS

VARIABLE PARAMETER	CYCLE COUNT	AVERAGE CYCLE TIME	AVERAGE NUMBER IN QUEUE	AVERAGE QUANTUM	AVERAGE OVERHEAD	COMPUTATIONAL EFFICIENCY	AVERAGE EXCHANGE OVERHEAD	AVERAGE EXCHANGE TIME
1.0000	5010.0000	2.8723	18.8168	.0903	.9854	59.1873	.0058	.1483
2.0000	4517.0000	3.1854	18.8027	.1046	.9945	61.7143	.0059	.1468
3.0000	5015.0000	2.8715	18.7460	.0906	.9827	59.1653	.0058	.1496

AVERAGE OVERLOAD	MAXIMUM QUANTUM	MAXIMUM CYCLE TIME	MAXIMUM NUMBER IN QUEUE	MAXIMUM OVERHEAD	MAXIMUM OVERLOAD	MAXIMUM NUMBER STATIONS	STATIONS SERVICED
23.2286	2.0000	4.7983	19.0000	1.0920	21.0000	40.0000	267.0000
13.6003	2.0000	5.1182	19.0000	1.0620	20.0000	40.0000	286.0000
23.2385	2.0000	3.5123	19.0000	1.0920	21.0000	40.0000	262.0000

FIGURE 3.4.12

SIMUALTOR OUTPUT DATA
WITH EARLY TERMINATIONS ALLOWED,
NOT ALLOWED AND BACKGROUND JOBS PERMITTED
QUANTUM DETERMINATION METHOD Q_4, LARGE JOB DATA, AND 40 STATIONS

VARIABLE PARAMETER	CYCLE COUNT	AVERAGE CYCLE TIME	AVERAGE NUMBER IN QUEUE	AVERAGE QUANTUM	AVERAGE OVERHEAD	COMPUTATIONAL EFFICIENCY	AVERAGE EXCHANGE OVERHEAD	AVERAGE EXCHANGE TIME
1.0000	5005.0000	2.8757	18.8905	.0896	.9916	58.8255	.0058	.1499
2.0000	4508.0000	3.1928	18.8791	.1038	1.0015	61.3546	.0059	.1480
3.0000	5019.0000	2.8692	18.8745	.0893	.9904	58.7254	.0058	.1517

AVERAGE OVERLOAD	MAXIMUM QUANTUM	MAXIMUM CYCLE TIME	MAXIMUM NUMBER IN QUEUE	MAXIMUM OVERHEAD	MAXIMUM OVERLOAD	MAXIMUM NUMBER STATIONS	STATIONS SERVICED
40.2659	2.0000	3.6155	19.0000	1.1220	31.0000	50.0000	263.0000
23.6399	2.0000	3.4435	19.0000	1.0620	31.0000	50.0000	282.0000
40.2638	2.0000	3.4663	19.0000	1.1020	31.0000	50.0000	259.0000

FIGURE 3.4.13

SIMUALTOR OUTPUT DATA
WITH EARLY TERMINATIONS ALLOWED,
NOT ALLOWED AND BACKGROUND JOBS PERMITTED
QUANTUM DETERMINATION METHOD Q_1, SCHOOL DATA, AND 50 STATIONS

VARIABLE PARAMETER	CYCLE COUNT	AVERAGE CYCLE TIME	AVERAGE NUMBER IN QUEUE	AVERAGE QUANTUM	AVERAGE OVERHEAD	COMPUTATIONAL EFFICIENCY	AVERAGE EXCHANGE OVERHEAD	AVERAGE EXCHANGE TIME
1.0000	2755.0000	4.2310	3.2526	.7537	.2870	57.9428	.0015	.2648
2.0000	2302.0000	5.0573	3.2354	.8509	.3026	54.4399	.0016	.2878
3.0000	6265.0000	2.2985	3.7168	.5099	.2073	82.4474	.0011	.1848

AVERAGE OVERLOAD	MAXIMUM QUANTUM	MAXIMUM CYCLE TIME	MAXIMUM NUMBER IN QUEUE	MAXIMUM OVERHEAD	MAXIMUM OVERLOAD	MAXIMUM NUMBER STATIONS	STATIONS SERVICED
.0047	2.0000	6.9366	9.0000	.5020	1.0000	10.0000	294.0000
.0039	2.0000	7.0178	9.0000	.4620	2.0000	10.0000	297.0000
.0010	2.0000	4.1620	8.0000	.4420	1.0000	10.0000	212.0000

FIGURE 3.4.14

SIMUALTOR OUTPUT DATA
WITH EARLY TERMINATIONS ALLOWED, NOT ALLOWED AND BACKGROUND JOBS PERMITTED
QUANTUM DETERMINATION METHOD Q_4, SCHOOL DATA, AND 50 STATIONS

VARIABLE PARAMETER	CYCLE COUNT	AVERAGE CYCLE TIME	AVERAGE NUMBER IN QUEUE	AVERAGE QUANTUM	AVERAGE OVERHEAD	COMPUTATIONAL EFFICIENCY	AVERAGE EXCHANGE OVERHEAD	AVERAGE EXCHANGE TIME
1.0000	3973.0000	3.6193	18.7475	.0825	.9771	42.7221	.0057	1.0600
2.0000	3402.0000	4.2261	18.6864	.1050	1.0038	46.4396	.0059	1.1360
3.0000	4140.0000	3.4790	18.6181	.0781	.9643	41.7832	.0057	1.0361

AVERAGE OVERLOAD	MAXIMUM QUANTUM	MAXIMUM CYCLE TIME	MAXIMUM NUMBER IN QUEUE	MAXIMUM OVERHEAD	MAXIMUM OVERLOAD	MAXIMUM NUMBER STATIONS	STATIONS SERVICED
9.7224	2.0000	5.2822	19.0000	1.0820	10.0000	30.0000	174.0000
5.0091	2.0000	5.3609	19.0000	1.0620	11.0000	30.0000	193.0000
9.8123	2.0000	5.0269	19.0000	1.0720	11.0000	30.0000	164.0000

FIGURE 3.4.15

SIMUALTOR OUTPUT DATA
WITH EARLY TERMINATIONS ALLOWED,
NOT ALLOWED AND BACKGROUND JOBS PERMITTED
QUANTUM DETERMINATION METHOD Q_1, LARGE JOB DATA, AND 50 STATIONS

VARIABLE PARAMETER	CYCLE COUNT	AVERAGE CYCLE TIME	AVERAGE NUMBER IN QUEUE	AVERAGE QUANTUM	AVERAGE OVERHEAD	COMPUTATIONAL EFFICIENCY	AVERAGE EXCHANGE OVERHEAD	AVERAGE EXCHANGE TIME
1.0000	4090.0000	3.5190	18.7910	.0778	.9755	41.5563	.0057	1.0534
2.0000	3413.0000	4.2166	18.7823	.1044	1.0094	46.5198	.0059	1.1344
3.0000	4081.0000	3.5289	18.7283	.0780	.9774	41.4126	.0057	1.0557

AVERAGE OVERLOAD	MAXIMUM QUANTUM	MAXIMUM CYCLE TIME	MAXIMUM NUMBER IN QUEUE	MAXIMUM OVERHEAD	MAXIMUM OVERLOAD	MAXIMUM NUMBER STATIONS	STATIONS SERVICED
26.9340	2.0000	5.2727	19.0000	1.0820	21.0000	40.0000	160.0000
15.0674	2.0000	5.4828	19.0000	1.1120	20.0000	40.0000	189.0000
26.8508	2.0000	5.0849	19.0000	1.0920	20.0000	40.0000	159.0000

FIGURE 3.4.16

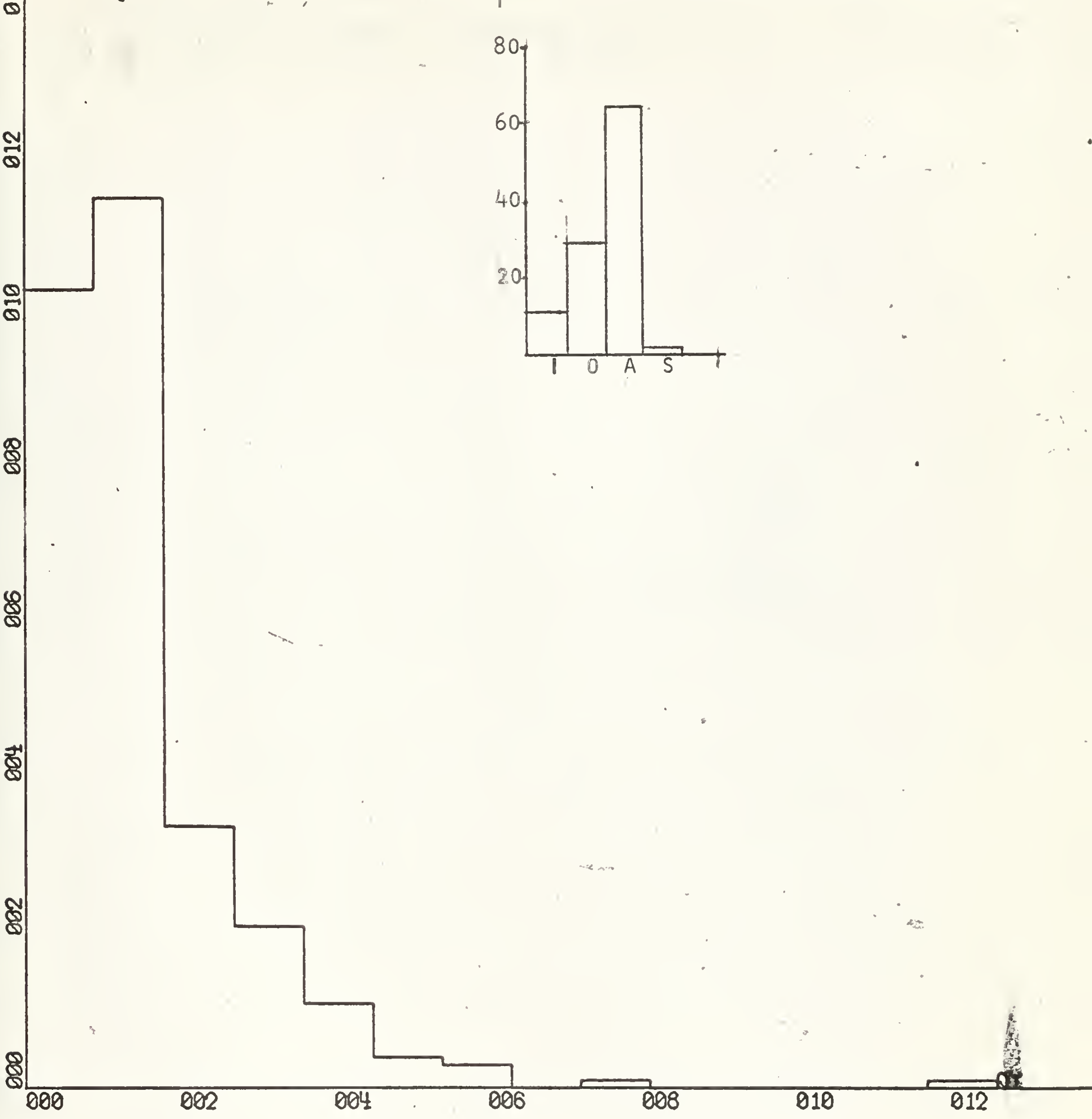
SIMUALTOR OUTPUT DATA
WITH EARLY TERMINATIONS ALLOWED,
NOT ALLOWED AND BACKGROUND JOBS PERMITTED
QUANTUM DETERMINATION METHOD Q_4, LARGE JOB DATA, AND 50 STATIONS

VARIABLE PARAMETER	CYCLE COUNT	AVERAGE CYCLE TIME	AVERAGE NUMBER IN QUEUE	AVERAGE QUANTUM	AVERAGE OVERHEAD	COMPUTATIONAL EFFICIENCY	AVERAGE EXCHANGE OVERHEAD	AVERAGE EXCHANGE TIME
1.0000	4126.0000	3.4881	18.8769	.0774	.9982	41.8684	.0057	1.0122
2.0000	3473.0000	4.1437	18.8779	.1034	1.0069	47.1075	.0058	1.1189
3.0000	4091.0000	3.5204	18.8614	.0764	1.0008	40.9183	.0057	1.0637

AVERAGE OVERLOAD	MAXIMUM QUANTUM	MAXIMUM CYCLE TIME	MAXIMUM NUMBER IN QUEUE	MAXIMUM OVERHEAD	MAXIMUM OVERLOAD	MAXIMUM NUMBER STATIONS	STATIONS SERVICED
42.8536	2.0000	5.2955	19.0000	1.1020	31.0000	50.0000	158.0000
24.3281	2.0000	5.6533	19.0000	1.1620	31.0000	50.0000	190.0000
42.4749	2.0000	5.3750	19.0000	1.1220	30.0000	50.0000	158.0000

FIGURE 3.4.17

QUANTUM DETERMINATION METHOD Q_1, SCHOOL DATA, AND 10 STATIONS



The step interval for abscissa for this and all other graphs in this section is 90 seconds.

X-SCALE = 2.00E+02 UNITS/INCH.
Y-SCALE = 2.00E+01 UNITS/INCH.

HISTOGRAM OF TOTAL TIME IN SYSTEM
OTTO/GRIMES THESIS

FIGURE 3.4.18

QUANTUM DETERMINATION METHOD Q_4, SCHOOL DATA, AND 10 STATIONS

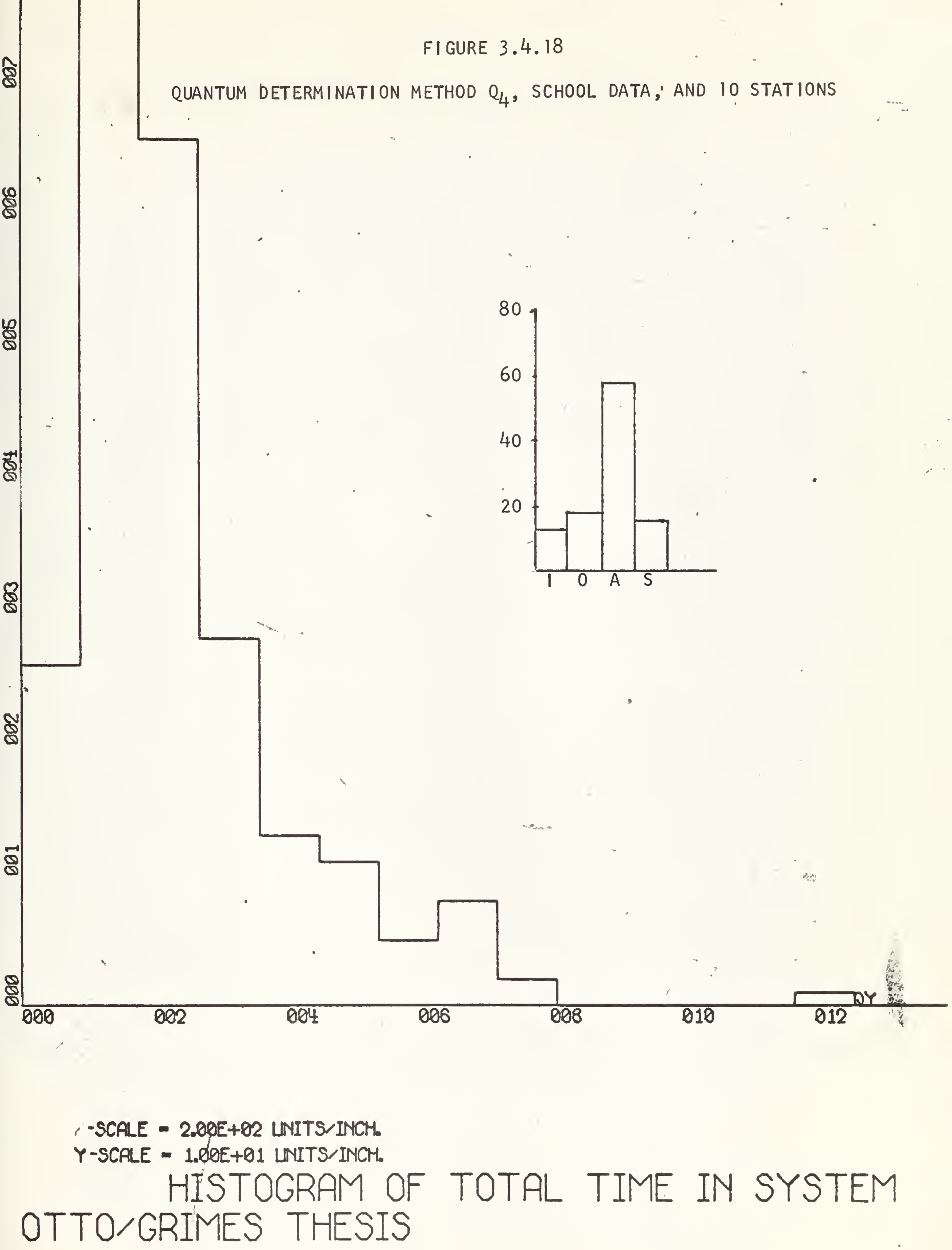
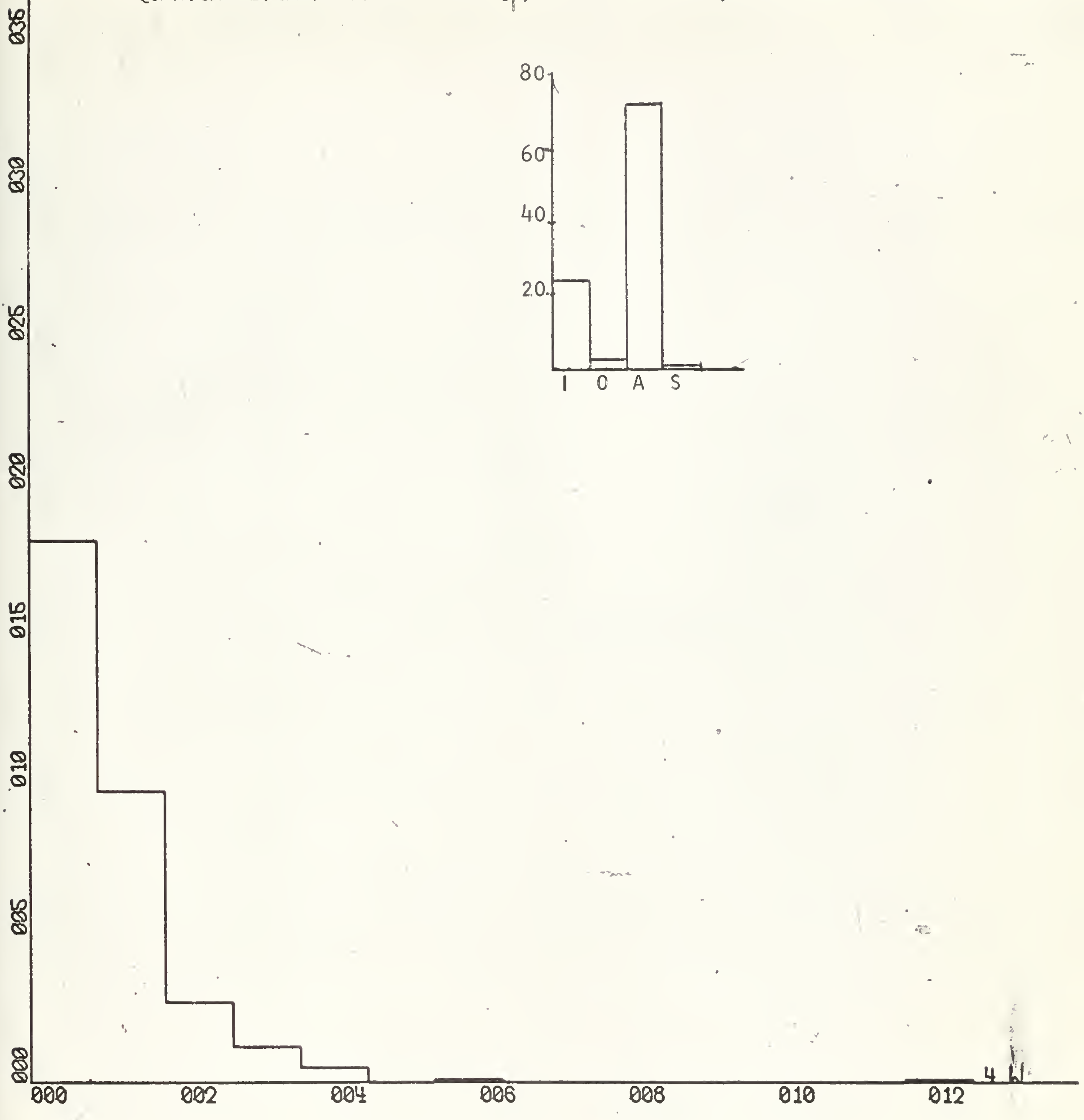


FIGURE 3.4.19

QUANTUM DETERMINATION METHOD Q_1, LARGE JOB DATA, AND 10 STATIONS

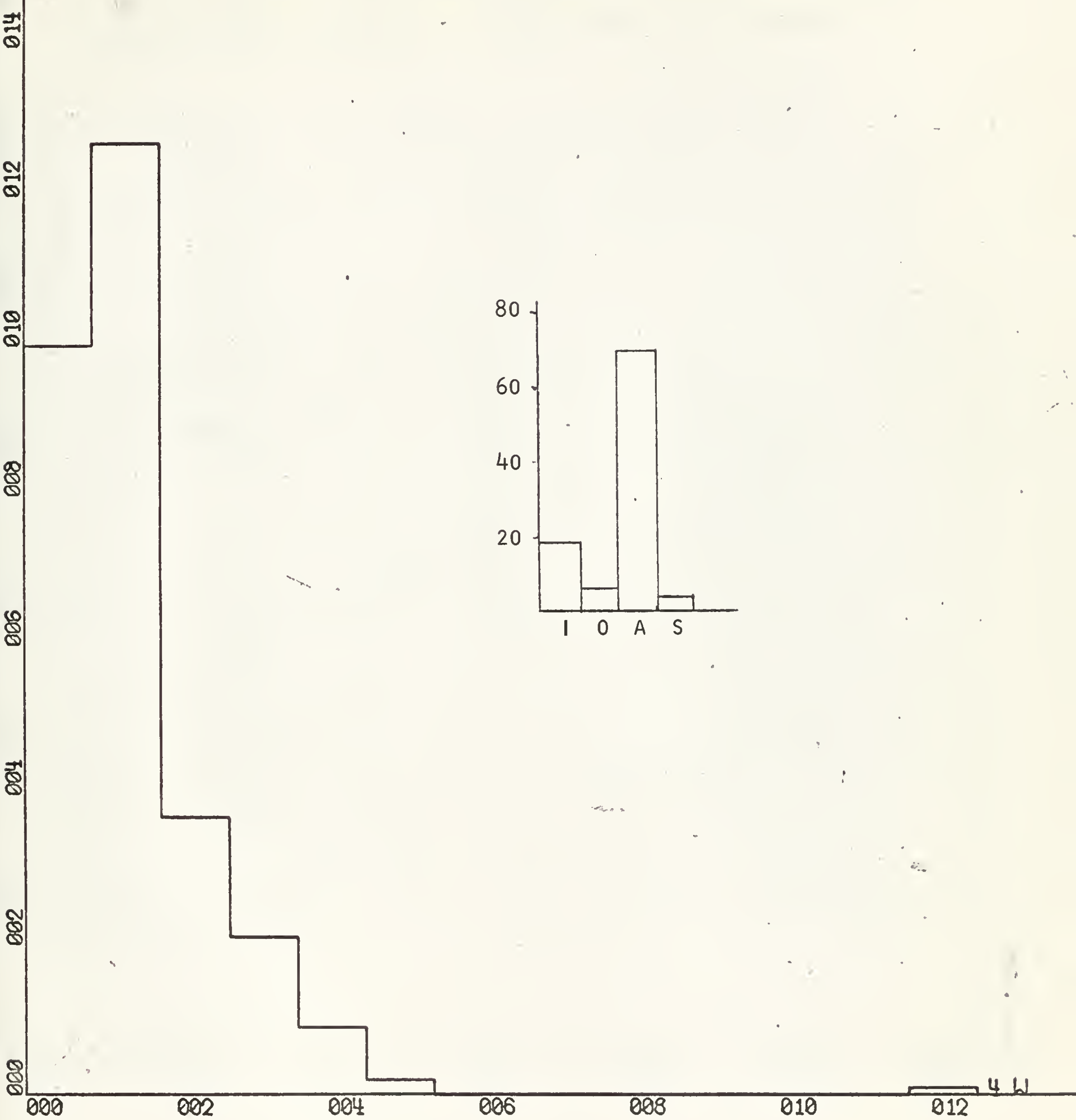


X-SCALE = 2.00E+02 UNITS/INCH.
Y-SCALE = 5.00E+01 UNITS/INCH.

HISTOGRAM OF TOTAL TIME IN SYSTEM
OTTO/GRIMES THESIS

FIGURE 3.4.20

QUANTUM DETERMINATION METHOD Q_4, LARGE JOB DATA, AND 10 STATIONS

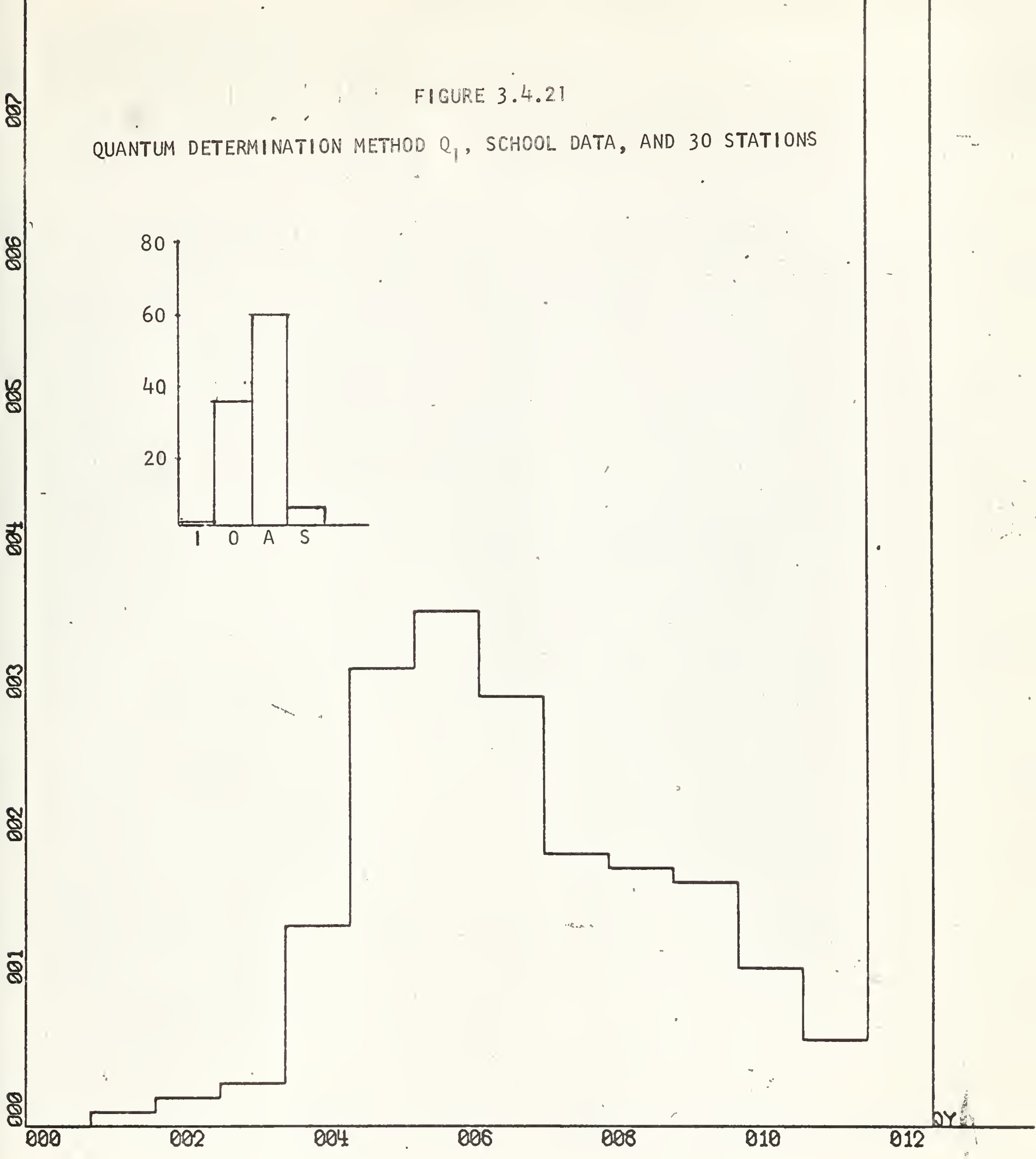


X-SCALE = 2.00E+02 UNITS/INCH.
Y-SCALE = 2.00E+01 UNITS/INCH.

HISTOGRAM OF TOTAL TIME IN SYSTEM
OTTO/GRIMES THESIS

FIGURE 3.4.21

QUANTUM DETERMINATION METHOD Q_1, SCHOOL DATA, AND 30 STATIONS



X-SCALE = 2.00E+02 UNITS/INCH.
Y-SCALE = 1.00E+01 UNITS/INCH.

HISTOGRAM OF TOTAL TIME IN SYSTEM
OTTO/GRIMES THESIS

FIGURE 3.4.22

QUANTUM DETERMINATION METHOD Q_4, SCHOOL DATA, AND 30 STATIONS

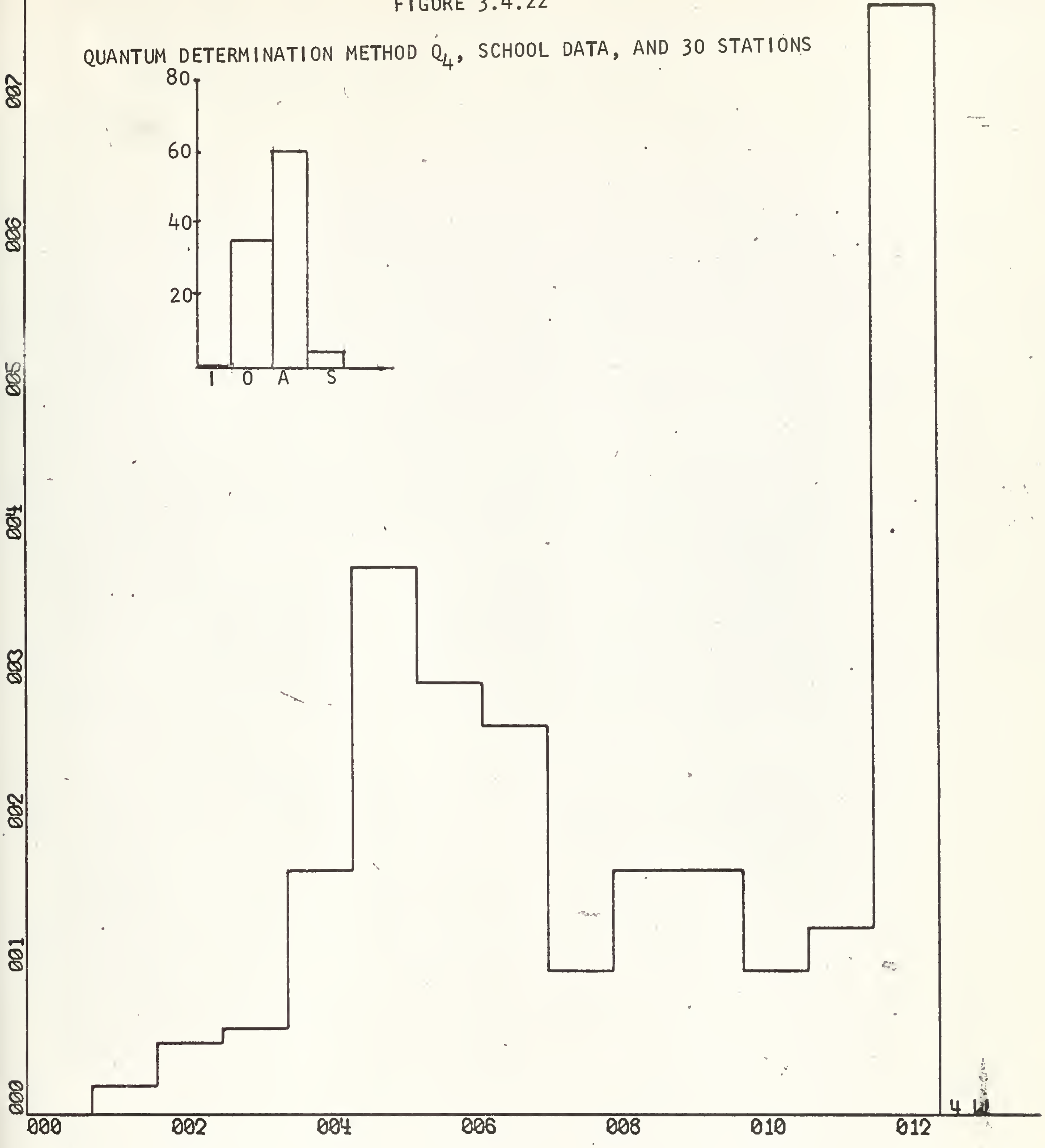
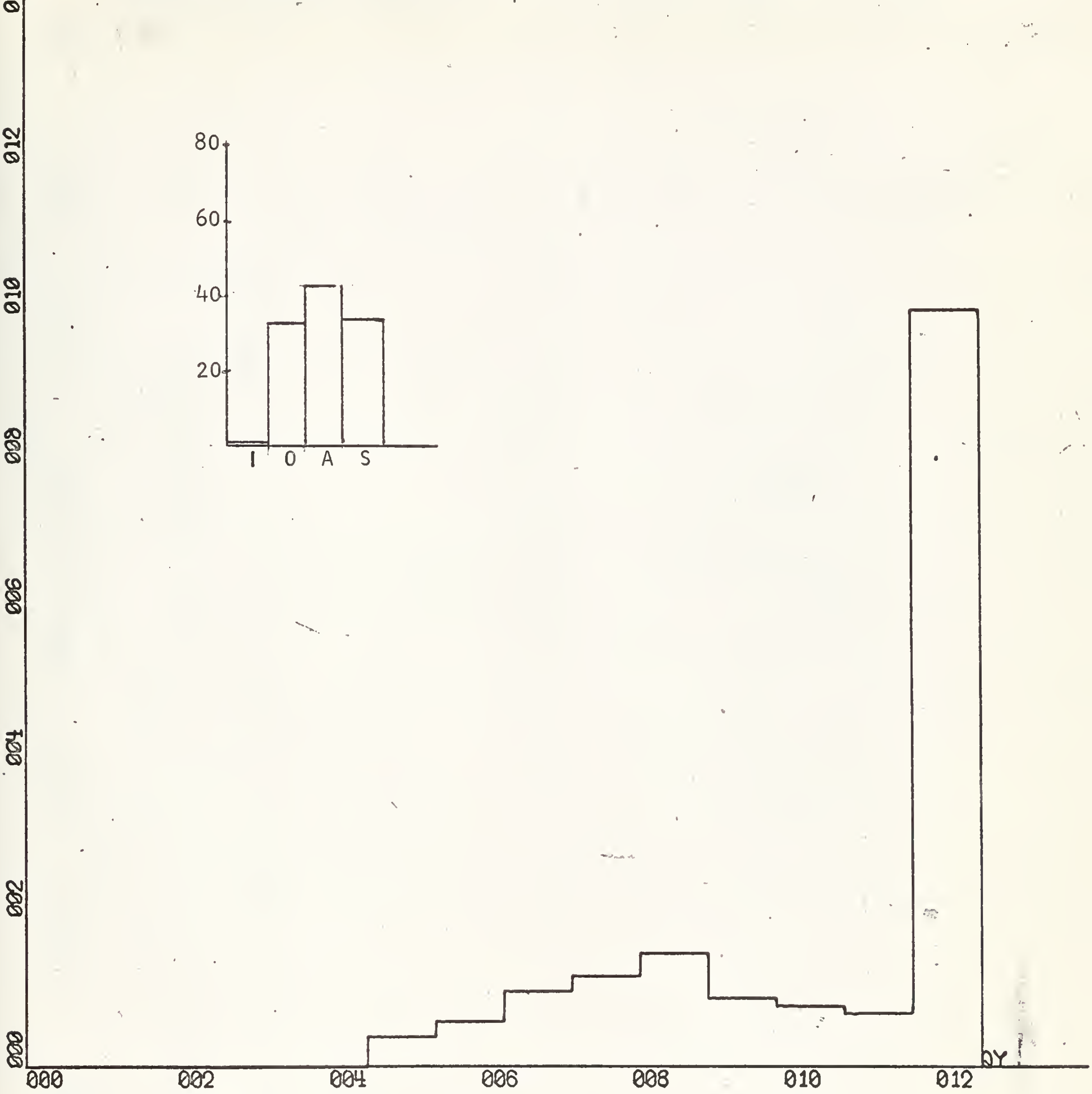


FIGURE 3.4.23

QUANTUM DETERMINATION METHOD Q_1, LARGE JOB DATA, AND 30 STATIONS



X-SCALE = 2.00E+02 UNITS/INCH.
Y-SCALE = 2.00E+01 UNITS/INCH.

HISTOGRAM OF TOTAL TIME IN SYSTEM
OTTO/GRIMES THESIS

FIGURE 3.4.24

QUANTUM DETERMINATION METHOD Q_4, LARGE JOB DATA, AND 30 STATIONS

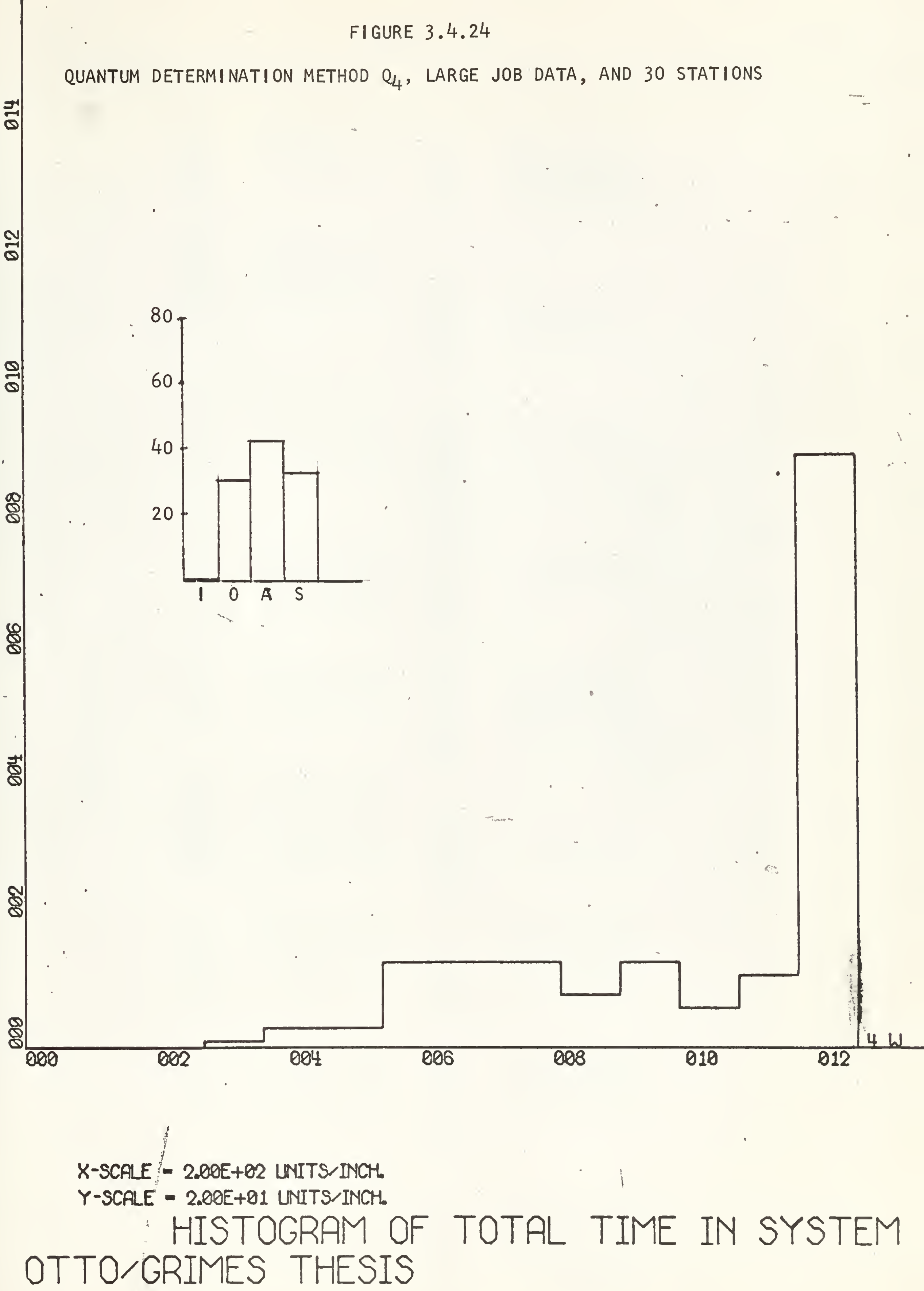


FIGURE 3.5.1

SIMULATOR OUTPUT DATA

```
VARIABLE     CYCLE    AVERAGE   AVERAGE    AVERAGE  AVERAGE  COMPUTATIONAL AVERAGE  AVERAGE
PARAMETER    COUNT     CYCLE    NUMBER     QUANTUM  OVERHEAD EFFICIENCY    EXCHANGE EXCHANGE
                       TIME     IN QUEUE                                   OVERHEAD TIME

    1        15200     .8548     2.6718     .190     .1599      63.58       .0009    .1467
    1         4900    2.9342    13.1131     .105     .7030      48.70       .0041    .7910
    1         4159    3.4571    19.1479     .068    1.0081      38.86       .0059   1.0968
    1         4146    3.4711    19.4262     .068    1.0187      38.73       .0060   1.0993
    1         4106    3.5053    19.7991     .067    1.0459      38.70       .0061   1.0936
    1         3586    4.0091    23.0312     .063    1.2057      36.92       .0071   1.3139
    1         3452    4.1689    24.0385     .061    1.2528      36.07       .0074   1.4027
    1         3414    4.2161    24.6169     .061    1.2913      36.11       .0075   1.3919

AVERAGE    MAXIMUM   MAXIMUM   MAXIMUM   MAXIMUM   MAXIMUM   MAXIMUM   STATIONS    AVERAGE
OVERLOAD   QUANTUM   CYCLE     NUMBER    OVERHEAD  OVERLOAD  NUMBER    SERVICED    SERVICE
                     TIME      IN QUEUE                      STATIONS              TIME

  .0003     .2000    2.9593       9        .4920      1        10        236       238.44
  .0014     .2000    4.5362      20       1.0220      1        20        193      1001.22
 8.4720     .2000    4.7277      20       1.1320     10        30        149      1997.41
24.6345     .2000    4.8024      20       1.1220     20        40        147      2025.50
40.8639     .2000    4.7661      20       1.1720     29        50        148      2081.36
 1.5367     .2000    5.4749      25       1.3720      5        30        141      2363.31
17.5032     .2000    5.9064      25       1.3820     15        40        133      2497.99
34.6295     .2000    5.4520      25       1.3920     25        50        136      2427.64
```

LARGE JOB DATA POLICY NO. 1

QUANTUM DETERMINATION METHOD ONE

VARIABLE PARAMETERS - MAXIMUM QUEUE AND NUMBER OF STATIONS

FIGURE 3.5.2

SIMULATOR OUTPUT DATA

VARIABLE PARAMETER	CYCLE COUNT	AVERAGE CYCLE TIME	AVERAGE NUMBER IN QUEUE	AVERAGE QUANTUM	AVERAGE OVERHEAD	COMPUTATIONAL EFFICIENCY	AVERAGE EXCHANGE OVERHEAD	AVERAGE EXCHANGE TIME
1	3539	4.0626	24.1568	.056	1.2565	33.93	.0074	1.4200
1	3204	4.4916	28.6217	.045	1.4840	29.43	.0088	1.6794
1	3204	4.4920	29.4254	.045	1.5335	29.53	.0089	1.6254
1	2844	5.0605	32.8762	.043	1.6984	28.74	.0100	1.9008
1	2811	5.1209	34.0352	.043	1.7705	28.41	.0102	1.8892
1	2750	5.2327	34.3065	.042	1.7739	28.12	.0105	1.9811
1	2733	5.2664	38.0648	.031	1.9705	22.22	.0115	2.1229
1	2596	5.5454	40.3598	.029	2.0855	21.31	.0121	2.2745

AVERAGE OVERLOAD	MAXIMUM QUANTUM	MAXIMUM CYCLE TIME	MAXIMUM NUMBER IN QUEUE	MAXIMUM OVERHEAD	MAXIMUM OVERLOAD	MAXIMUM NUMBER STATIONS	STATIONS SERVICED	AVERAGE SERVICE TIME
.0020	.2000	5.5291	30	1.5420	1	30	131	2597.18
10.3630	.2000	5.9384	30	1.6220	10	40	102	3488.90
25.2192	.2000	5.8438	30	1.6720	20	50	108	3843.78
2.6850	.2000	6.4293	35	1.8820	5	40	104	4028.84
17.3110	.2000	6.7006	35	1.9120	14	50	103	3992.35
.0364	.2000	6.7402	40	2.0420	1	40	99	4159.53
11.1285	.2000	6.8026	40	2.1420	12	50	72	5205.38
7.4570	.2000	6.8329	48	2.4820	11	50	74	5853.14

LARGE JOB DATA POLICY NO. 1

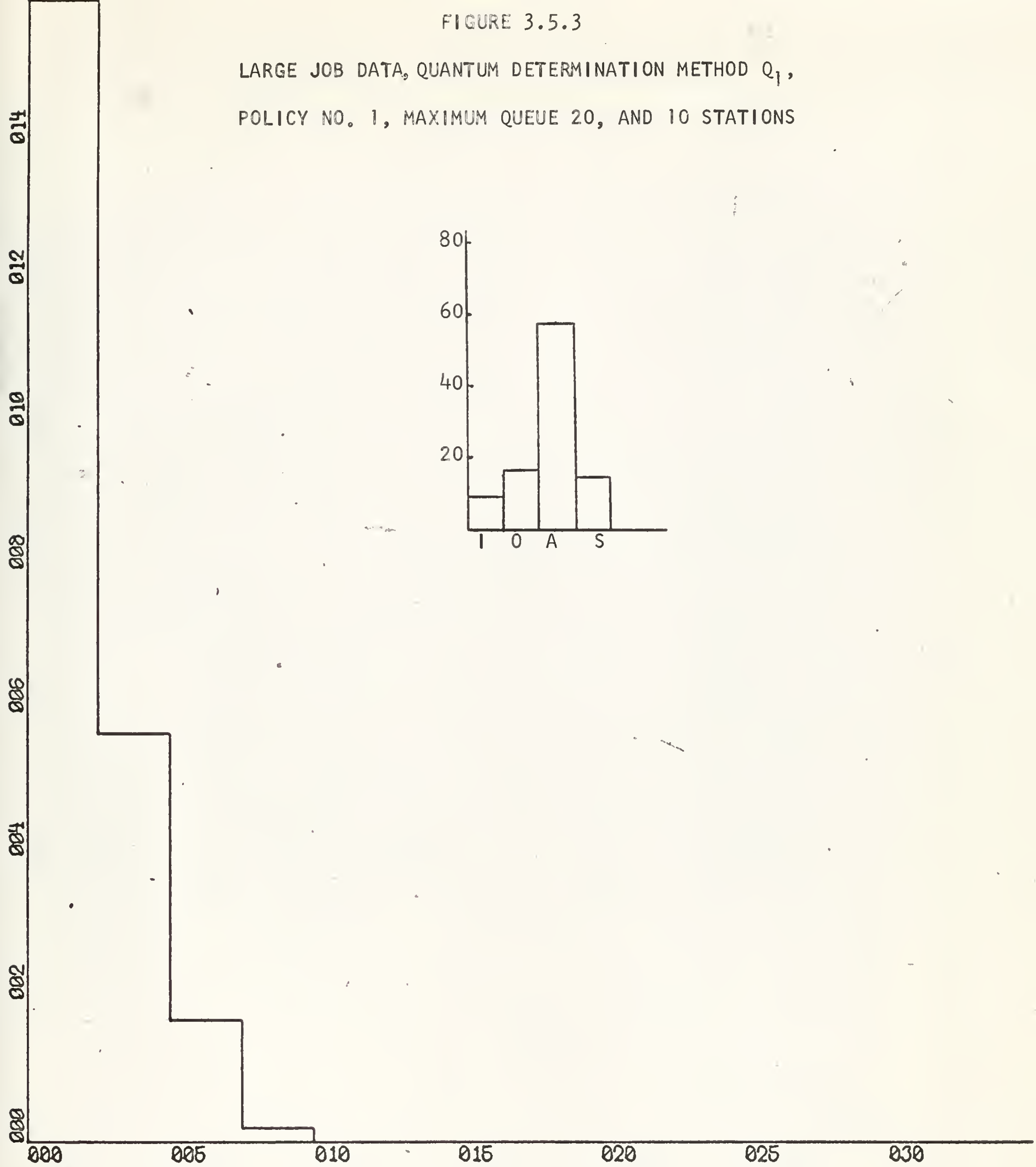
QUANTUM DETERMINATION METHOD ONE

VARIABLE PARAMETERS - MAXIMUM QUEUE AND NUMBER OF STATIONS

FIGURE 3.5.3

LARGE JOB DATA, QUANTUM DETERMINATION METHOD Q_1,

POLICY NO. 1, MAXIMUM QUEUE 20, AND 10 STATIONS



Interval size equals 250 seconds.

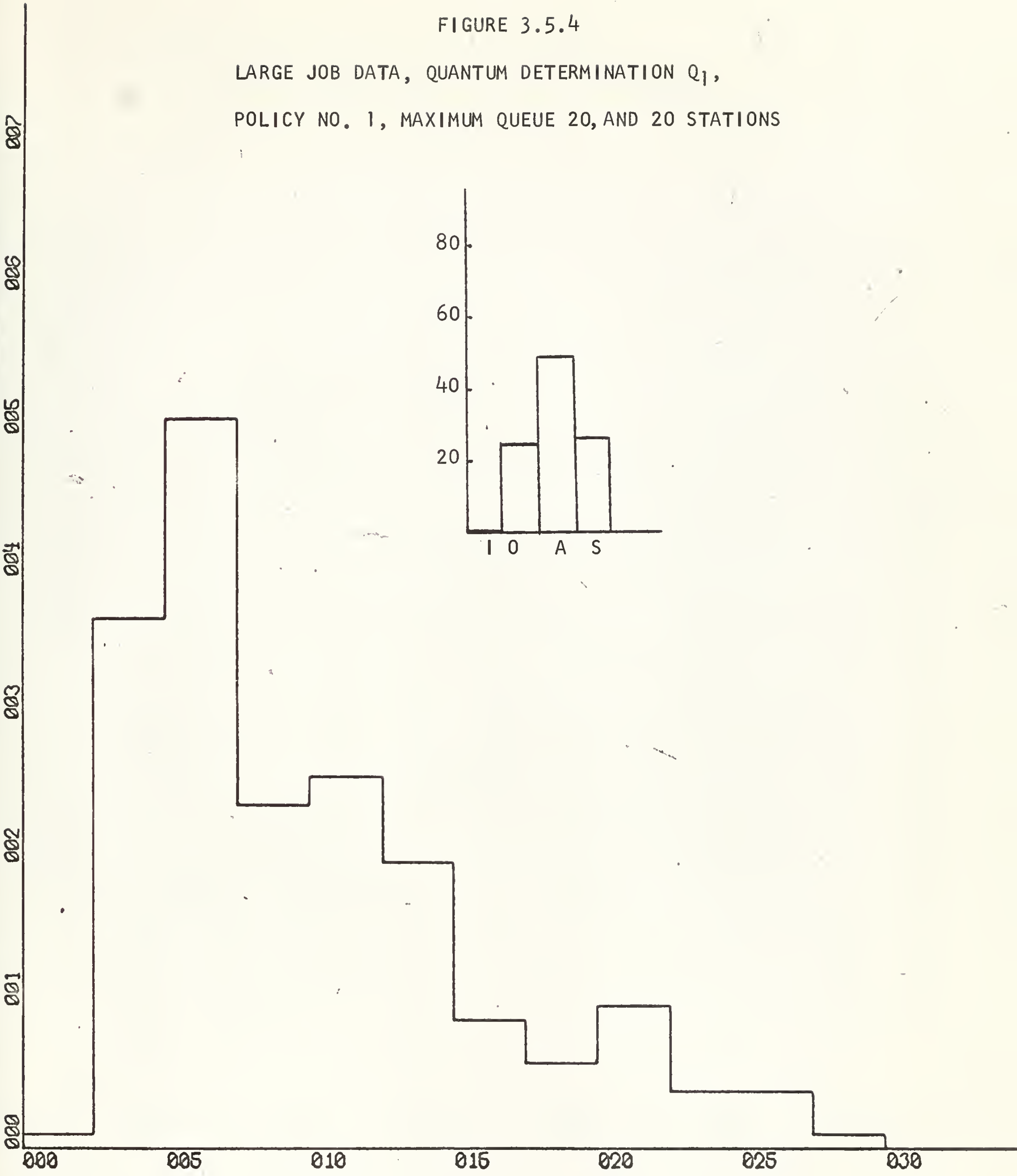
X-SCALE = 5.00E+02 UNITS/INCH.
Y-SCALE = 2.00E+01 UNITS/INCH.

HISTOGRAM OF TOTAL TIME IN SYSTEM
OTTO/GRIMES THESIS

FIGURE 3.5.4

LARGE JOB DATA, QUANTUM DETERMINATION Q_1,

POLICY NO. 1, MAXIMUM QUEUE 20, AND 20 STATIONS



Interval size equals 250 seconds.

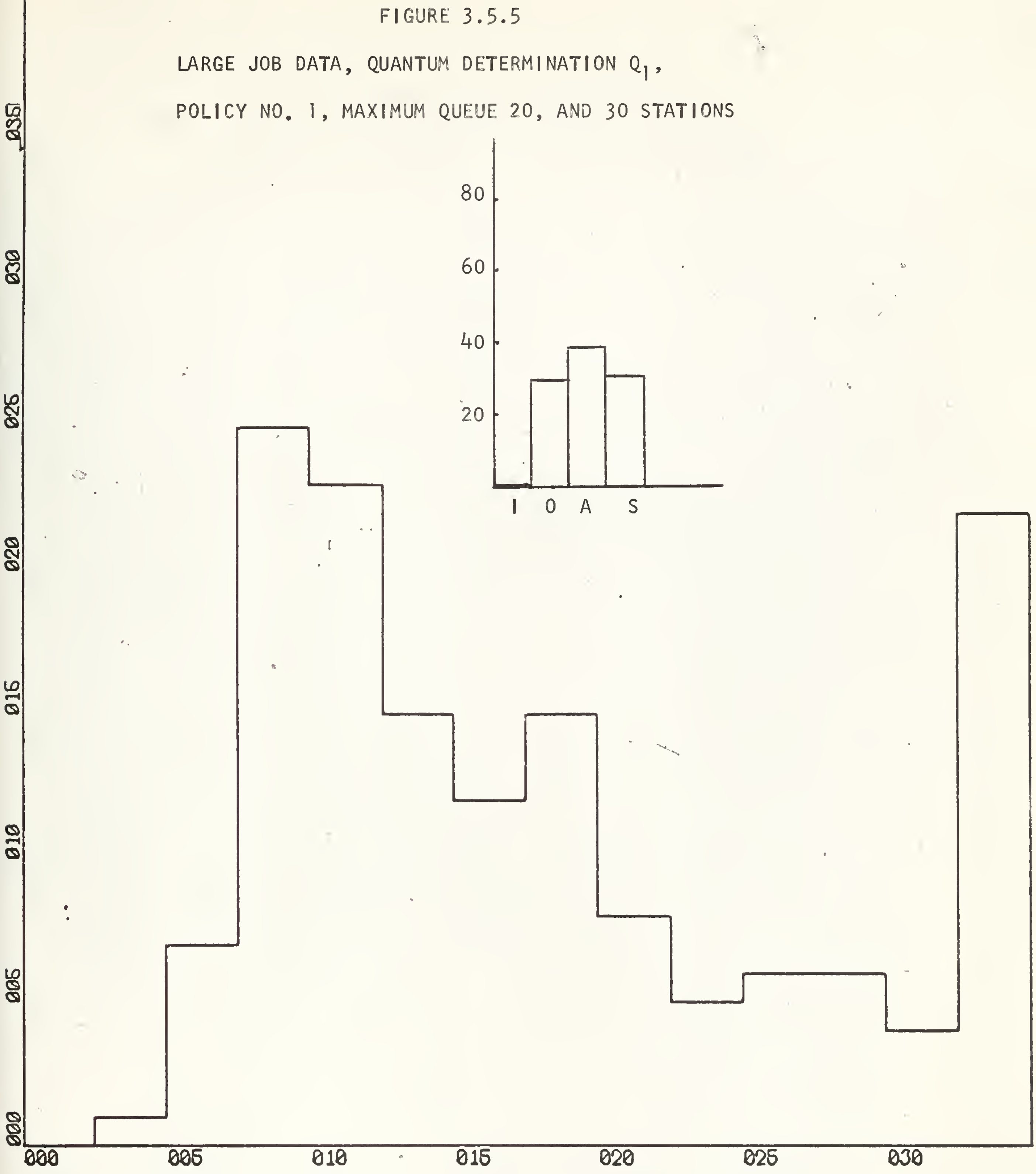
X-SCALE = 5.00E+02 UNITS/INCH.
Y-SCALE = 1.00E+01 UNITS/INCH.

HISTOGRAM OF TOTAL TIME IN SYSTEM
OTTO/GRIMES THESIS

FIGURE 3.5.5

LARGE JOB DATA, QUANTUM DETERMINATION Q_1,

POLICY NO. 1, MAXIMUM QUEUE 20, AND 30 STATIONS



Interval size equals 250 seconds.

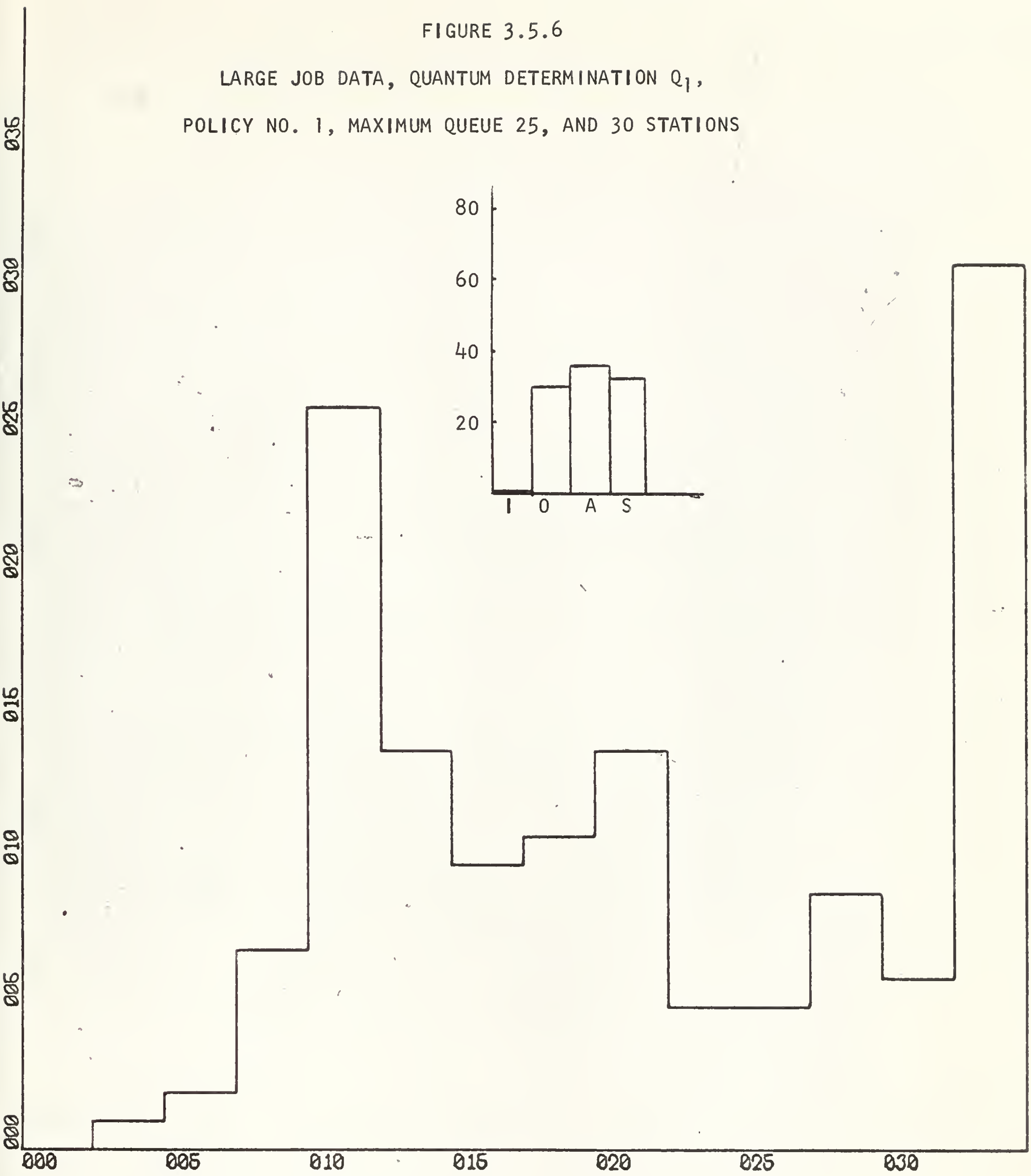
X-SCALE = 5.00E+02 UNITS/INCH.
Y-SCALE = 5.00E+00 UNITS/INCH.

HISTOGRAM OF TOTAL TIME IN SYSTEM
OTTO/GRIMES THESIS

FIGURE 3.5.6

LARGE JOB DATA, QUANTUM DETERMINATION Q_1,

POLICY NO. 1, MAXIMUM QUEUE 25, AND 30 STATIONS



Interval size equals 250 seconds.

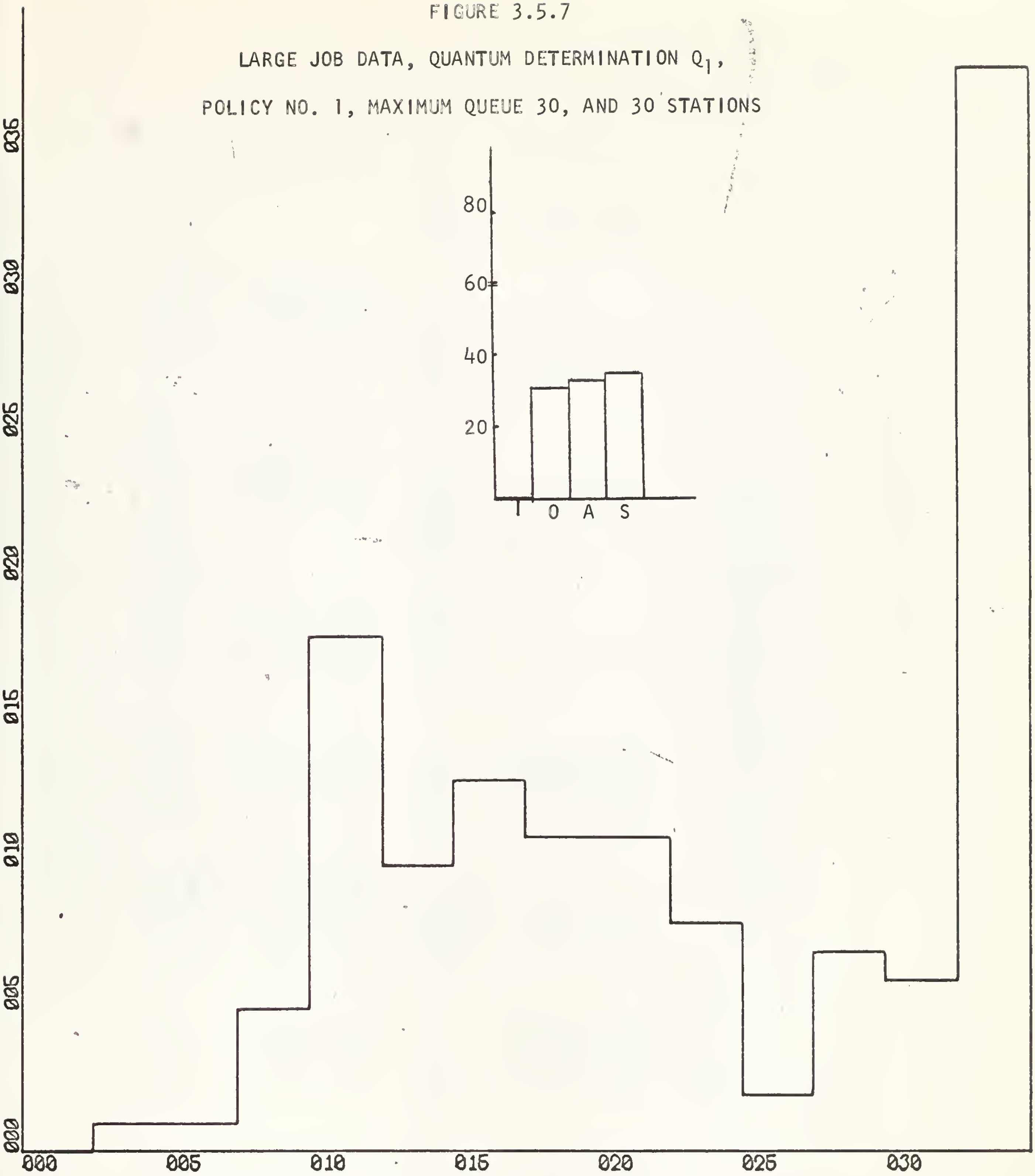
X-SCALE = 5.00E+02 UNITS/INCH.
Y-SCALE = 5.00E+00 UNITS/INCH.

HISTOGRAM OF TOTAL TIME IN SYSTEM
OTTO/GRIMES THESIS

FIGURE 3.5.7

LARGE JOB DATA, QUANTUM DETERMINATION Q_1,

POLICY NO. 1, MAXIMUM QUEUE 30, AND 30 STATIONS



Interval size equals 250 seconds.

X-SCALE = 5.00E+02 UNITS/INCH.
Y-SCALE = 5.00E+00 UNITS/INCH.

HISTOGRAM OF TOTAL TIME IN SYSTEM
OTTO/GRIMES THESIS

FIGURE 3.5.8

SIMULATOR OUTPUT DATA

```
VARIABLE      CYCLE    AVERAGE    AVERAGE    AVERAGE  AVERAGE   COMPUTATIONAL AVERAGE   AVERAGE
PARAMETER     COUNT     CYCLE     NUMBER     QUANTUM  OVERHEAD  EFFICIENCY    EXCHANGE  EXCHANGE
                        TIME      IN QUEUE                                    OVERHEAU  TIME

    1         20608     .5995      2.2141     .192     .1331       75.09       .0007     .0148
    1          6682    2.1494     10.7389     .131     .5844       68.62       .0034     .0861
    1          5527    2.6015     19.0190     .074     .9983       55.63       .0059     .1542
    1          5482    2.6249     19.4301     .072    1.0204       55.08       .0060     .1569
    1          5392    2.6692     19.7932     .072    1.0408       54.95       .0061     .1598
    1          5006    2.8719     21.7435     .070    1.1377       54.22       .0067     .1755
    1          4694    3.0654     24.0643     .066    1.2526       52.80       .0074     .1928
    1          4631    3.1081     24.6171     .065    1.2792       52.71       .0075     .1892

AVERAGE     MAXIMUM   MAXIMUM    MAXIMUM    MAXIMUM   MAXIMUM   MAXIMUM     STATIONS     AVERAGE
OVERLOAD    QUANTUM   CYCLE      NUMBER     OVERHEAD  OVERLOAD  NUMBER      SERVICED     SERVICE
                      TIME       IN QUEUE                       STATIONS                 TIME

   .0002     .2000    2.3267        9         .4920      1        10          295         149.44
   .0012     .2000    3.0605       19         .9920      1        20          314         558.32
  5.2696     .2000    3.4478       20        1.1320      9        30          253        1212.04
 20.3426     .2000    3.3960       20        1.1320     19        40          246        1244.63
 36.5639     .2000    3.4198       20        1.1320     29        50          244        1276.66
   .4989     .2000    3.6634       25        1.3420      4        30          244        1265.41
 13.4649     .2000    3.7195       25        1.4120     14        40          235        1526.15
 30.0298     .2000    3.6468       25        1.3820     23        50          233        1545.43
```

SCHOOL DATA POLICY NO. 1

QUANTUM DETERMINATION METHOD ONE

VARIABLE PARAMETERS - MAXIMUM QUEUE AND NUMBER OF STATIONS

```
                                FIGURE 3.5.9

                           SIMULATOR OUTPUT DATA

VARIABLE       CYCLE    AVERAGE    AVERAGE    AVERAGE  AVERAGE   COMPUTATIONAL  AVERAGE   AVERAGE
PARAMETER      COUNT     CYCLE     NUMBER     QUANTUM  OVERHEAD  EFFICIENCY     EXCHANGE  EXCHANGE
                         TIME      IN QUEUE                                     OVERHEAU  TIME

       1       5000     2.8752    22.1892      .067    1.1611      53.37         .0068     .1785
       1       4648     3.0960    28.6885      .048    1.4825      44.83         .0088     .2268
       1       4554     3.1607    29.4477      .047    1.5225      44.45         .0090     .2342
       1       4276     3.3651    31.8267      .046    1.6399      44.00         .0097     .2461
       1       4052     3.5519    34.1602      .044    1.7576      43.37         .0104     .2553
       1       4216     3.4131    32.4352      .046    1.6684      43.91         .0099     .2478
       1       4413     3.2613    38.9773      .026    1.9880      31.70         .0118     .2449
       1       3910     3.6813    44.0951      .026    2.2515      31.52         .0134     .2756

AVERAGE    MAXIMUM    MAXIMUM   MAXIMUM    MAXIMUM   MAXIMUM   MAXIMUM    STATIONS   AVERAGE
OVERLOAD   QUANTUM    CYCLE     NUMBER     OVERHEAD  OVERLOAD  NUMBER     SERVICED   SERVICE
                      TIME      IN QUEUE                       STATIONS              TIME

   .0044    .2000     3.7467      29        1.4920       1        30         238     1283.50
  6.7309    .2000     3.5829      30        1.6720       9        40         197     1986.85
 21.7324    .2000     3.6090      30        1.6420      19        50         194     2033.96
  1.0381    .2000     4.1349      35        1.8520       5        40         195     2121.79
 14.5094    .2000     4.1572      35        1.8820      15        50         191     2295.69
   .0050    .2000     4.3130      39        1.9820       1        40         192     2054.49
  7.8486    .2000     4.3430      40        2.1420      10        50         138     3238.06
   .0036    .2000     4.2400      50        2.5320       1        50         139     3645.17

            SCHOOL DATA                                    POLICY NO. 1
                       QUANTUM DETERMINATION METHOD ONE
            VARIABLE PARAMETERS - MAXIMUM QUEUE AND NUMBER OF STATIONS
```

thesG833
A study of some software parameters in t
3 2768 002 13938 8
DUDLEY KNOX LIBRARY